Lab 5

-Korodi Andreea-Cristina-
-Georgina Loba-

Git link: https://github.com/GeorgianaLoba/FLCD---but-teamed

Assignment for a team of 2 students!

Statement: Implement a parser algorithm

One of the following parsing methods will be chosen (assigned by teaching staff):

1.a. recursive descendent

1.b. ll(1)

1.c. lr(0)

The representation of the parsing tree (output) will be (decided by the team):

2.a. productions string (max grade = 8.5)

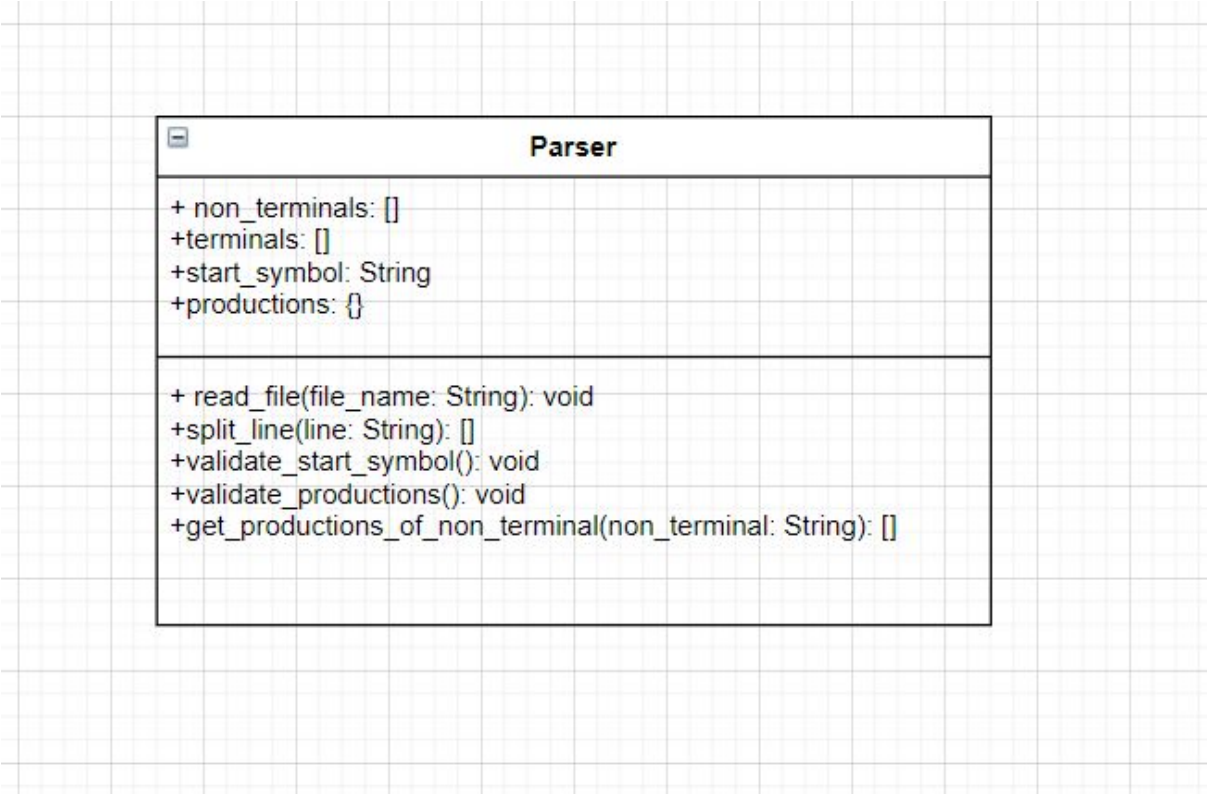
2.b. derivations string (max grade = 9)

2.c. table (using father and sibling relation) (max grade = 10)

PART 1: Deliverables

1. **Class grammar (required operations: read a grammar from file, print set of nonterminals, set of terminals, set of productions, production for a given nonterminal)**
2. Input file: g1.txt (grammar from seminar); g2.txt (grammar of the minilanguage; syntax rules from Lab1)
3. Functions corresponding to parsing strategy (see table below)

Class diagram:



File.in :

```
S Aa
a b
S
S - a Aa
A - a Aa | b Aa | a | b
```

File.in has the following structure:
First line: non_terminals
Second line: terminals
Third line: start_symbol
Starting from fourth line we have the productions.

Test:

```
def main():
    parser = Parser()
    parser.read_file('file.in')
    try:
        parser.validate_start_symbol()
        parser.validate_productions()
    except Exception as ex:
        print(ex)
        return
```