[Former-languages-and-compiler-design/Lab5 at main · 915-Nistor-Anca/Former-languages-and-compiler-design (github.com)](https://github.com/915-Nistor-Anca/Former-languages-and-compiler-design/tree/main/Lab5)

**Class Grammar:**

*\_\_init\_\_(self, file\_name):* Initializes the Grammar class with the name of the file the grammar is written in, and also with an empty list for the terminals, non-terminals, production rules and the start symbol.

*readFromFile(self):* Reads the grammar from the given file. On the first line, there are the non-terminals, on the second one, the terminals, on the third there is the start symbol and from the fifth line there are the production rules. The function splits the non-terminals and terminals by comma. It adds everything from the file in the Grammar class.

*\_\_str\_\_(self):* Returns a string with evertyhing the class has as attributes, just like it is in the file.

*breakProductionRule(self, production\_rule):* Breaks the given production in two parts, one with the left-hand side (the non-terminal) and the other with what comes after ->.

*checkIfCFG(self):* Checks and returns a message whether the given grammar is a Context Free Grammar or not. Being a CFG means one production rule has to start with the start symbol, the left hand side of each production rule has to be one non-terminal and everything from the right hand side can be found on the non-terminals or terminals list.