

<https://github.com/Phe0niX12/FTCD/tree/main/lab5/LFTClab5>

The Grammar class represents a context-free grammar (CFG) and provides methods for initializing, reading CFG information from a file, checking CFG validity, and obtaining a string representation of the grammar.

Attributes:

EPSILON (class attribute): A string constant representing the empty string, often denoted as  $\epsilon$  or "epsilon."

non\_term (list): Non-terminals of the CFG.

term (list): Terminals of the CFG.

start (string): The starting symbol or axiom of the CFG.

production (dictionary): Stores the finite set of productions, where keys are non-terminals, and values are lists of possible right-hand sides.

Methods: `__init__(self)` Initializes a Grammar object with empty lists for non-terminals, terminals, an empty string for the starting symbol, and an empty production dictionary.

`rebuild(self)` Resets all attributes of the Grammar object to an empty state, facilitating the reconstruction of a new CFG .

`__processLine(self, line: str, delimiter=' ')` -> List[str] Helper method that processes a line of text by splitting it into elements based on a specified delimiter. Returns a list of elements.

`read_from_file(self, file_name: str)` -> None Reads CFG information from the specified file, populating the Grammar object with non-terminals, terminals, the starting symbol, and production rules.

`check_cfg(self)` -> bool Checks the validity of the CFG by ensuring it has a starting symbol, and all symbols used in production rules are either non-terminals, terminals, or epsilon. Returns True if valid, False otherwise.

`__str__(self)` -> str Generates a string representation of the CFG, including non-terminals, terminals, the starting symbol, and production rules. Suitable for displaying the CFG details when printed.