https://github.com/cs-ubbcluj-ro/lab-work-computer-science-2024-TeodoraVlad12/tree/main/3-Parser

**Grammar Class Documentation**

**Attributes**

1. **nonterminals**: A list of all nonterminal symbols in the grammar.
2. **terminals**: A list of all terminal symbols in the grammar.
3. **productions**: A dictionary where:
   * Keys are nonterminal symbols.
   * Values are lists of production rules (each production rule is a string).
4. **start\_symbol**: The starting symbol of the grammar.

**Methods**

1. **\_\_init\_\_()**
   * Initializes the grammar with empty lists for nonterminals and terminals, an empty dictionary for productions, and None for the start symbol.
2. **read\_from\_file(file\_path)**
   * Reads a grammar definition from a file.
   * **File Format**:
     1. First line: Space-separated list of nonterminal symbols.
     2. Second line: Space-separated list of terminal symbols.
     3. Third line: The start symbol.
     4. Subsequent lines: Production rules in the format Nonterminal -> rule1 | rule2.
   * Updates the nonterminals, terminals, start\_symbol, and productions attributes.
3. **print\_nonterminals()**
   * Prints all nonterminal symbols in the grammar.
4. **print\_terminals()**
   * Prints all terminal symbols in the grammar.
5. **print\_productions()**
   * Prints all production rules in the grammar in the format Nonterminal -> rule.
6. **print\_productions\_for\_nonterminal(nonterminal)**
   * Prints all production rules for a specific nonterminal.
   * If the nonterminal does not exist, it prints an appropriate message.
7. **is\_cfg()**
   * Validates whether the grammar is a proper context-free grammar (CFG). It checks:
     1. The start symbol is defined and is a valid nonterminal.
     2. There is no overlap between terminal and nonterminal symbols (they must be distinct).
     3. Every production rule is valid:
        + The left-hand side of a rule must be a single nonterminal.
        + The right-hand side must consist only of terminals, nonterminals, or be empty (epsilon).
   * Returns True if the grammar is valid, otherwise prints errors and returns False.
8. **display\_menu()**
   * Displays an interactive menu for grammar operations. Allows the user to:
     1. Print nonterminals.
     2. Print terminals.
     3. Print all production rules.
     4. Print production rules for a specific nonterminal.
     5. Check if the grammar is a CFG.
     6. Exit the menu.