Github Link: https://github.com/Striletchi-Vlad/FLCD-L1
Grammar Class
Description

The Grammar class represents a context-free grammar (CFG) and provides methods for reading a grammar from a file, processing its productions, and performing various operations related to the grammar.
Class Attributes

nonterminals: A set containing the nonterminal symbols of the grammar.
terminals: A set containing the terminal symbols of the grammar.
productions: A list containing tuples representing productions in the form (nonterminal, rhs_symbol).
start_symbol: The start symbol of the grammar.

Methods
__init__(self) -> None

Constructor method that initializes the Grammar object with empty sets for nonterminals and terminals, an empty list for productions, and a None start symbol.

read_grammar_from_file(self, filename: str) -> None

Reads a CFG from a file specified by filename. Each line in the file is expected to represent a production.

process_production(self, production: str) -> None

Processes a single production, extracting nonterminals, terminals, and adding the production to the list.

process_rhs_symbol(self, nonterminal: str, rhs_symbol: str) -> None

Processes the right-hand side (rhs) of a production, adding the production to the list and updating the set of terminals.

print_nonterminals(self) -> None

Prints the nonterminal symbols of the grammar.

print_terminals(self) -> None

Prints the terminal symbols of the grammar.

print_productions(self) -> None

Prints all productions in the grammar.

production
s_for_nonterminal(self, nonterminal: str) -> List[str]

Returns a list of right-hand side symbols for a given nonterminal.

is_cfg(self) -> bool

Checks if the grammar is a context-free grammar (CFG) by verifying that each nonterminal in a production is also in the set of nonterminals.

Example Usage

python