Lab 6

Cupes Andrei and Cotor Catinca-Florina

https://github.com/andreicupes/FLCD/tree/main/Lab5

Created the Parser Class.

The generateFirst function generates the first set for a given grammar. The first set is a collection of terminal symbols that can appear at the beginning of a production generated by a given nonterminal symbol. The function works as follows:

- Initialize the first set by adding all terminal symbols that can appear at the beginning of any production that starts with a nonterminal.
- Continue iterating until the first set no longer changes:
    - Create a new map to store the updated first set.
    - For each nonterminal in the grammar:
        - Retrieve the set of productions that can be generated by the nonterminal symbol.
        - Take the union of the current first set and the set of terminal symbols that can appear at the beginning of each production, after removing any nonterminal symbols from the beginning of the production.
        - If the first set for the nonterminal has changed, update the first set and set the isChanged flag to true.
- Return the final first set.

The function uses the concatenationOfSizeOne method to concatenate the first sets of the nonterminal symbols on the right-hand side of a production with the first terminal symbol on the right-hand side of the production. This is used to update the first set for each nonterminal in the grammar.

The generateFollow function is a method used in an LL(1) parser to generate the follow set for each non-terminal symbol in the grammar. The follow set is a set of terminal symbols that can appear immediately after a given non-terminal in a valid sentence of the language described by the grammar.

The function first initializes an empty set of follow symbols for each non-terminal in the grammar. It then adds the special "epsilon" symbol (which represents the empty string) to the follow set of the starting symbol of the grammar.

Next, the function enters a loop where it calculates new follow sets for each non-terminal by iterating over all productions in the grammar. For each production, it checks if the non-terminal appears in the right-hand side of the production. If it does, the function uses the first set of the symbol immediately following the non-terminal in the production to update the follow set for the non-terminal.

The function continues iterating over the productions until the follow sets stop changing, at which point the follow sets have been fully generated. The generated follow sets are then stored in the followSet instance variable.