Github link:

https://github.com/AndreiCotor/FLCD-Team-project
https://github.com/AndreiCotor/FLCD-Team-project/pull/2/files

Function read_pif()

The purpose of this function is to read and process data from the files "token.in" and "pif.out" to generate a list of processed tokens.

Function get_parsing_table(grammar)

The purpose is to construct the LR Parsing Table which is represented as a list of rows(which is a class containing the information for a row) tailored for a given grammar. The parsing table is systematically populated by incorporating shift and reduce actions, reflecting the transitions between states in the canonical collection. Shift actions are determined based on the lookahead symbols, while reduce actions are assigned according to the grammar's production rules. The resulting parsing table encapsulates the necessary information for LR parsing, enabling the parsing algorithm to make informed decisions during the parsing process.

Function parse(list_of_strings, grammar):

The parse function takes a list of strings and an LR grammar as input, performing LR parsing to generate a parsing tree. It maintains two stacks, the working stack and the remaining stack, and employs a parsing table obtained from the grammar. The function iteratively processes the input strings, executing shift, reduce, or accept actions based on the LR parsing algorithm. Exception handling is integrated, covering scenarios like encountering invalid states, attempting shifts with an empty remaining stack, facing invalid symbols during goto operations, encountering unrecognized action types, and ensuring the presence of an ACCEPT action before the stacks become empty. The final output is the parsing tree resulting from the parsing process, providing insights into the syntactic structure of the input strings based on the given grammar.

Class ParesOutput

The class serves as a representation of the output produced by an LR parser, encapsulating information about the parsing tree. It maintains a list of ParsingTreeRow instances, each embodying details such as index, info, parent, and right_sibling. The class offers methods to append a ParsingTreeRow to the list, print the formatted parsing tree to the console, and write it to a specified file.