Burak Tekin
3203697
Parsing

README

General Informations

Programming Language: Python
Version: 3.6.5
itertools library used for obtaining cartesian products in rule productions, os library used

Module, functions, classes

There is only one class called 'Parser'. it has 11 methods inside which are respectively:

- **__init__:** initializing the class and its variables
- **find_terminals:** it finds out the terminal symbols by checking the difference between RHS and LHS.
- **lhs_rhs:** Creates a list of symbols which are place RHS of the rule and the LHS of the rule.
- **load_grammar:** reads the grammar file and splitting of RHS and LHS happens here.
- **find_lhs:** finds all LHS symbols of a given symbol which occurs on the RHS.
- **new_productions:** Implemented to find new rules after the epsilon-product removed.
- **is_start_symbol_in_rhs:** finds S if occurs on the RHS and return a boolean to create new production rule for it.
- **is_epsilon_exist:** looks for "epsilon" productions and tries to remove them out and calls new_productions to create new rules. (STILL AN ON GOING WORK)
- **is_unit_production_exist:** Checks for productions has only 1 symbol length and tries to fix them for CNF.
- **is_long_productions_exist:** this looks for the rules have RHS longer than or equal to 2 symbols and tries to fix them for CNF.
- **converter:** applying CNF conversions over the grammar to make it ready to apply CYK parser.
- **cyk_parser:** Runs the parser to create parsing tree. (It does not work properly, so I left it as it is, needed more time to make it work)

Guidance

- Open the project folder
- Open a terminal window and type: "*source venv/bin/activate*" and press enter
- Type the following command: *pip3 install -r requirements.txt*
- Type *python3 parser.py*