Github link: https://github.com/RazvanAndreiLazar/FLCD

LR(0) grammar
+ ParsingOutput – table with parent and sibling
- Works for g2.txt + PIF.out

Classes:
Production:
- lhs – left hand side of a production
- rhs – right hand side of a production
-

Grammar:
- terminals – list with all the terminals
- non-terminals – list with all the non-terminals
- initial – initial state
- production_dict – dictionary of productions (lhs → Production)
- production_list – list with all the productions
- Methods:
  - productions_for_one – returns all the productions of the given non_terminal
  - read – reads the grammar from a file

Action (Enum): SHIFT, ACCEPT, REDUCE

Parsing table
- table – the parsing table (list(dict (action: Action, reduction: production_no, goto: set (state_no))))
- Methods:
  - add_set – adds a new set of states to the parsing table
  - process_canonical_collection – add all the sets in a canonical collection to the parsing table
  - get_action_for_set – get the action given a set_no
  - get_goto_destination – get the destination given the set number and term
  - get_reduction – get the production for the reduction given the set number
  - get_productions_numbering – returns a list with the productions (indexes are used as numbers for the reductions)

State
- prod – production of the state
- index – the index of the point (.)
- string_after_point – the first term after the point
- Methods:
  - shift_dot_right – shifts the it index one terminal to the right

Node
- child – the child of the node (Node)
- right_sibling – the right sibling of the node (Node)
- value – the information stored in the node
- depth – depth in the tree

ParsingOutput

- head – the head of the tree
- Methods:
  - search_parent – searches the rightmost node with the given value, that has no children, starting from the given node
  - add_production – adds a new production, the **lhs** is considered the father (the head of the tree if there is none, otherwise the search_parent(head, lhs)), the **rhs** is split in terms, that are added as children (the first child) or right_sibling (the rest of the children)
  - process_parser_output – gets a list of productions and creates the representation
  - print_to_file – prints the ParsingOutput to a file

Parser

- grammar – the grammar of the language
- pt – ParsingTable
- Methods:
  - closure – computes the closure of a set
  - goto – computes the closure of the set formed by all the states in the original set that have the first string after the point the given term, with the point shifted after it
  - canonical_collection – computes the canonical collection of the grammar
  - parse – parses the given array, returns the ParsingOutput if valid, otherwise throws error

PifReader

- Methods:
  - readPIF – reads the Program Internal Form from the file
  - get_keys – gets a list with only the keys (tokens) in the PIF