https://github.com/danielmalancioiu/LFTC

LR:

In our LR class we have the grammar, the working grammar and the ordered productions.

The working grammar is the enriched version of our grammar and we use it for computing the canonical collection of states.

The ordered productions, represent a List with pairs, that contain all the productions, placed separately (So if we had A-> a | ab they would be stored as A -> a, A -> ab)

In this class, we have all the computations for the LR(0) algorithm.

The methods:

public String getNonTerminalPrecededByDot(Item item)

public State closure(Item item)

public State goTo(State state, String elem)

public CanonicalCollection canonicalCollection(

public ParsingTable getParsingTable(CanonicalCollection canonicalCollection) throws Exception

public void parse(Stack<String> inputStack, ParsingTable parsingTable, String filePath) throws IOException

ParsingTreeRow:

This class is used to represend basically a node from the parsing tree which we are going to compute.

Here, for each node we store the index, the information which is basically the symbol (terminal or non-terminal), the parent, the right sibling, left child and level.

OutputTree:

The output tree class is used to compute the parse tree.

Here we have stored the root of the parse tree, the grammar, currentIndex, the indexInInput, the maximum level and the tree list.

The methods:

public ParsingTreeRow generateTreeFromSequence(List<Integer> inputSequence)

public ParsingTreeRow buildRecursive(int level, ParsingTreeRow parent, List<String> currentContent, List<Integer> inputSequence)

public void createList(ParsingTreeRow node)

CanonicalCollection

Fields:

-states - List that contains the states of the canonical collection

-adjacencyList - Map that keeps track of the states created by another states. In form of <Pair<Integer, String>, Integer>

Methods:

- conntectStates(Int, String, Int)

- addState(State)

ParsingTable

Fields:

- entries - List that contains elements of type RowTable

RowTable

Fields:

- stateIndex - integer containing the index of the state on the respective row

- action - a state action type containing the action of the state on the respectiv row

- reduceNonTerminal - a string containing the reduce non-termina of the state

- reduceContent - List of strings containing the reduce contents of the state

- shifts - List of pair string, integer containing the shifts of a state