Lab 7 Andrei Candet and Radu Ceaca

Link to git:

https://github.com/cinnamonbreakfast/flcd/tree/main/lab5_final_%40raduceaca

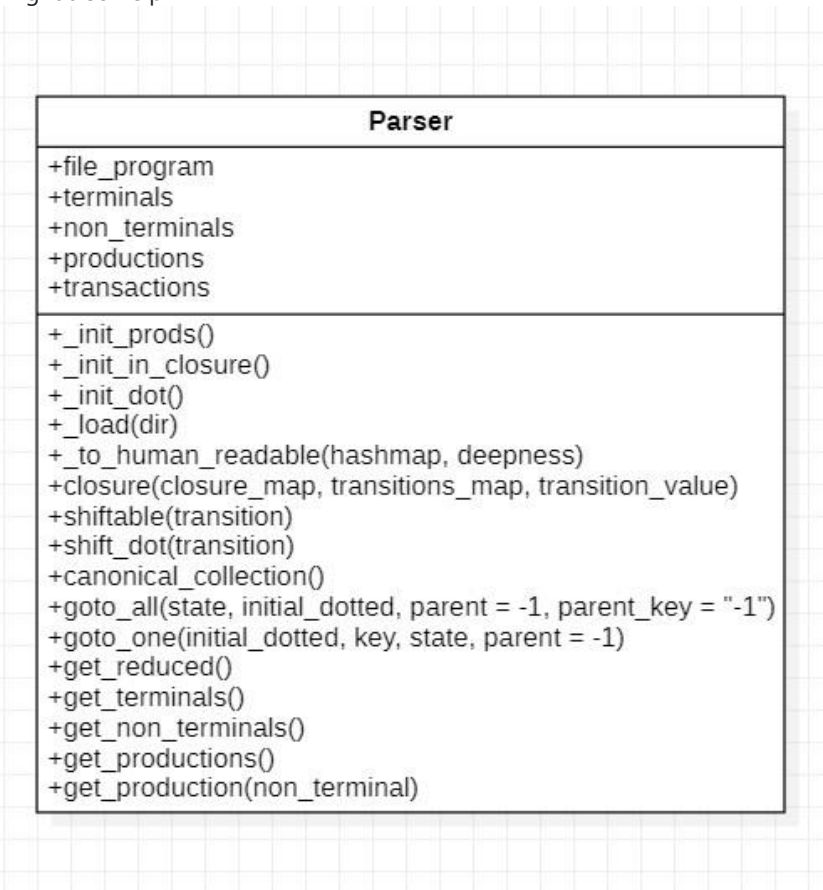
Assignment for a team of 2 students!

Statement: Implement a parser algorithm (cont)

PART 2: Deliverables

1. Algorithm corresponding to parsing tables (if needed) and parsing strategy
2. Class ParserOutput - DS and operations corresponding to choice 2.a/2.b/2.c (lab 5) (required operations: transform parsing tree into representation; print DS to screen and to file)

Remark:
- if the table contains conflicts, you will be helped to solve them. It is important to print message containing row (*symbol in LL(1), respectively state in LR(0)*) and column (symbol) where the conflict appears. For LL(1) values (αα,i) might also help



+ parse_string(string)

	action			Goto	
	a	b	$	A	S
0	S_3	S_4		2	1
1			acc		
2	S_3	S_4		6	
3	S_3	S_4		6	
4	r3	r3	r3		
5	r1	r1	r1		
6	r2	r2	r2		

_init_prods():

Initialize the production directory

_init_in_closure()

Initialeze the closure map

_init_dot ()

Initialization method for closure

_load(dir)

Load data from file closure(closure_map, transitions_map, transition_value) Compute the closure_map

shiftable(transition)

Check if the dot can be shifted

shift_dot(transition)

Call the shiftable function and shift the dot

canonical_collection()

Populate the canonical collection

goto_all(state, initial_dotted, parent=-1, parent_key="-1")

Goes through every state

goto_one(initial_dotted, key, state, parent=-1)

Goes to a single state get_reduced()

Returns the reduced map

get_terminals()

Returns terminals

get_non_terminals()

Returns non terminals

get_produtions()

Returns productions

get_production(non_terminal)

Returns the production of a non terminal

parse_string(string)

pre: string to parse

post: parsing table