#### **Lab 8 DOCUMENTATION**

https://github.com/lola23/LFTC/tree/master/Lab\_08/src

# <u>LL 1</u>

#### Fields:

grammar: Grammar

FIRST: map of non terminal with the non terminals that it has as first

FOLLOW: map of every non terminal and next symbol generated after it

numbererdProductions: map of all productions with an index

alpha: stack for terminals

beta: stack for nonterminals

pi: stack of indexes of productions

### parseTable()

PRE: -

POST: constructs the parsing table

## parse():

In: sequence to be checked

Out: true or false if sequence is accepted or not

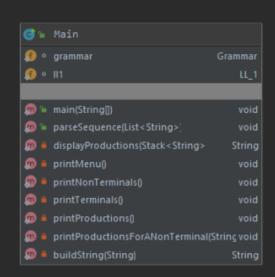
## numberProductions():

pre: -

post: numberedProductions is populated with every production and its



<b>G</b> •	Grammar	
<b>6</b>		List <string></string>
<b>6</b>		List <string></string>
<b>6</b>		String
<b>()</b>	P	Map < String, List < List < String > >:
@ °	readFromFile()	void
@ °	getN()	List <string></string>
0 -	getE()	List <string></string>
<b>@</b> %	getS()	String
@ °	getP()	Map < String, List < List < String > >:
0 1	getRhsProductions(String	Map < String, List < List < String > >:
<b>@</b> •	${\tt getProductionsInWhichNTIsOnTheRight(String)} \\$	Map < String, List < List < String > >:



P DS SL D BS AD TY IDL AD AA T AAL S SI ST AS E EX EX2 OP IS ST CS IFS EIFS ES C TRY EXT R LS **TREX** 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 P P -> 2 DS SL 3 $DS \rightarrow DDS \mid \sim$  $D \rightarrow BS \mid AD$ BS -> TY 0 IDL 23 TY -> 5 | 6 $IDL -> 240 IDL | \sim$ AD -> 7 19 TY 20 0 AA AAL 23  $AA \rightarrow 19 T 20 | \sim$ T -> 0 AA | 1 $AAL \rightarrow 240 AA AAL \mid \sim$  $SL \rightarrow SSL \mid \sim$  $S \rightarrow SI \mid ST$  $SI \rightarrow AS \mid IS$ AS -> 0.13 E 23E -> 16 EX | EX $EX \rightarrow T EX2$ EX2 -> OP T |  $\sim$ OP -> 31 | 32 | 33 | 34 | 35 | 14 | 15 IS -> 36 17 0 18 23 | 4 17 0 18 23  $ST \rightarrow CS \mid IFS \mid LS$ CS -> 21 SL 22IFS -> 8 C CS EIFS ES EIFS -> 9 C CS | ~  $ES -> 10 CS | \sim$ 

TRY -> 0 | 1

R -> 25 | 26 | 27 | 28 | 29 | 30 | 14 | 15

LS -> 11 17 5 0 23 E 12 E 23 1 18 21 S 22

EXT -> R TREX | OP TREX | ~

C -> 17 TREX 18

TREX -> TRY EXT

```
Example:
p1.in
begin_appy
      inty a, b, c, max;
      a is 10;
      b is 20;
      c is 12;
      ify ( a >= b and a >= c ) {
            max is a;
  } elseify (b \ge a and b \ge c) {
    max is b;
  } elsy {
    max is c;
  }
end_appy
p1PIF.out
2 5 0 24 0 24 0 24 0 23 0 13 1 23 0 13 1 23 0 13 1 23 8 17 0 25 0 14 0 25 0 18 21 0 13 0 23
22 9 17 0 25 0 14 0 25 0 18 21 0 13 0 23 22 10 21 0 13 0 23 22 3
=> Sequence is accepted
```

#### Code review:

https://github.com/eduardCeausoglu/FLCDLabs/commit/65 dae 928271b006 df 17092093080a5336c5ceb05