

Github link:

→ [https://github.com/Badetto/FLCD\\_Lab2](https://github.com/Badetto/FLCD_Lab2) (In the folder "Lab\_Teams")

Parser:

Description: The first step into creating a Parser algorithm with the parsing method LL(1) is creating the methods: FIRST and FOLLOW.

→ **FIRST(symbol: string):HashSet<string>**: It calculates the FIRST set for a given symbol in a context-free grammar, which includes all possible terminals that can appear at the start of strings derived from that symbol.

**Production Rules of Grammar**

$S \rightarrow ACB \mid Cbb \mid Ba$

$A \rightarrow da \mid BC$

$B \rightarrow g \mid \epsilon$

$C \rightarrow h \mid \epsilon$

**FIRST sets**

$$\begin{aligned} \text{FIRST}(S) &= \text{FIRST}(ACB) \cup \text{FIRST}(Cbb) \cup \text{FIRST}(Ba) \\ &= \{ d, g, h, b, a, \epsilon \} \end{aligned}$$

$$\begin{aligned} \text{FIRST}(A) &= \{ d \} \cup \text{FIRST}(BC) \\ &= \{ d, g, h, \epsilon \} \end{aligned}$$

$$\text{FIRST}(B) = \{ g, \epsilon \}$$

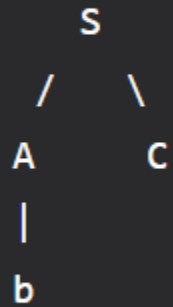
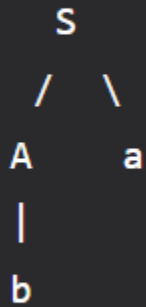
$$\text{FIRST}(C) = \{ h, \epsilon \}$$

→ **FOLLOW(nonTerminal: string, inProgress:HashSet<string> | null):HashSet<string>**: It computes the FOLLOW set for a given non-terminal in a context-free grammar, which includes all terminals that can appear immediately

after the non-terminal in any derivation from the start symbol.

$S \rightarrow Aa \mid Ac$

$A \rightarrow b$



Here,  $\text{FOLLOW}(A) = \{a, c\}$