**LAB SESSION 4: FIRST AND FOLLOW OF A CONTEXT FREE GRAMMAR**

**AIM**: To implement a program in python to compute the FIRST and FOLLOW of a given CFG.

**PROBLEM DEFINITION:** Develop a python program to compute the FIRST and FOLLOW of a given Context Free Grammar. Accept the Grammar from a file.

**THEORY:** The First and Follow sets are important in syntax analysis, mainly in parsing. All sets help in making predictive parsers and are integral to identifying how a given grammar can be parsed effectively.

The First Set for a non-terminal symbol represents all possible terminals that can appear at the beginning of any string derived from that non-terminal.

The follow set contains terminals that can appear immediately after a non-terminal in the derivation of the grammar.

All meaning of First and Follow sets is it allows to define which production rules are implemented in different parsing moment.

**What is the First Set?**

The First set of non-terminal in grammar tells us all the possible terminals that can appear at the beginning of any string derived from that non terminal. it gives us a way to know what symbol might come first if we start with a particular non-terminal. For example, if we have a rule like A -> aB | ε, then the First set of A includes the terminal a, as well as ε, which represents the possibility of deriving an empty string. Calculating the First set helps us predict which rules to apply during parsing by knowing which symbols can appear first in a derivation.

**What is a Follow Set?**

The Follow set of a non-terminal contains all terminals that can immediately follow it in any derivation of the grammar. This means that if a non-terminal appears in the middle of a sentence, the Follow set tells us what symbol might come right after it. For example, in a rule like S -> AB, the Follow set of A might include anything that can follow B, depending on other rules in the grammar. Follow sets are especially useful for parsers when making decisions about which rule to apply next and ensuring that we’re interpreting the structure of a sentence correctly, especially for LL(1) parsers where we only look one symbol ahead.

The follow set of the start symbol will always contain "$". Now the calculation of Follow falls under three broad cases:

If a Non-Terminal on the R.H.S. of any production is followed immediately by a Terminal then it can immediately be included in the Follow set of that Non-Terminal.

If a Non-Terminal on the R.H.S. of any production is followed immediately by a Non-Terminal, then the First Set of that new Non-Terminal gets included on the follow set of our original Non-Terminal. In case encountered an epsilon i.e. " # " then, move on to the next symbol in the production.

Note: "#" is never included in the Follow set of any Non-Terminal.

If reached the end of a production while calculating follow, then the Follow set of that non-terminal will include the Follow set of the Non-Terminal on the L.H.S. of that production. This can easily be implemented by recursion.

**PROGRAM**

**OUTPUT**

**CONCLUSION**