Name	Roll No.	Section	DoP
Vidhan Dahatkar	166	B	

Practical - 06

Aim- Design and Simulate Predictive / LL (1) Parsing Table using JFLAP for the grammar rules: A→aBa, B→bB.

Theory:

LL(1) Parsing:

LL(1) parsing is a top-down parsing technique that reads the input from Left to right (first L), constructs a Leftmost derivation (second L), and uses 1 lookahead symbol to make parsing decisions. It avoids backtracking by using a parsing table.

FIRST and FOLLOW Sets:

The FIRST set contains terminals that can appear at the beginning of strings derived from a symbol. The FOLLOW set contains terminals that can appear immediately to the right of a non-terminal in some derivation. These sets help determine entries in the LL(1) parsing table.

Predictive Parsing Table:

A predictive parsing table is a two-dimensional table indexed by non-terminals and input symbols (terminals). It tells the parser which production rule to apply based on the current non-terminal and the next input symbol. The table must not have conflicts to be considered LL(1).

Epsilon (ε) Productions:

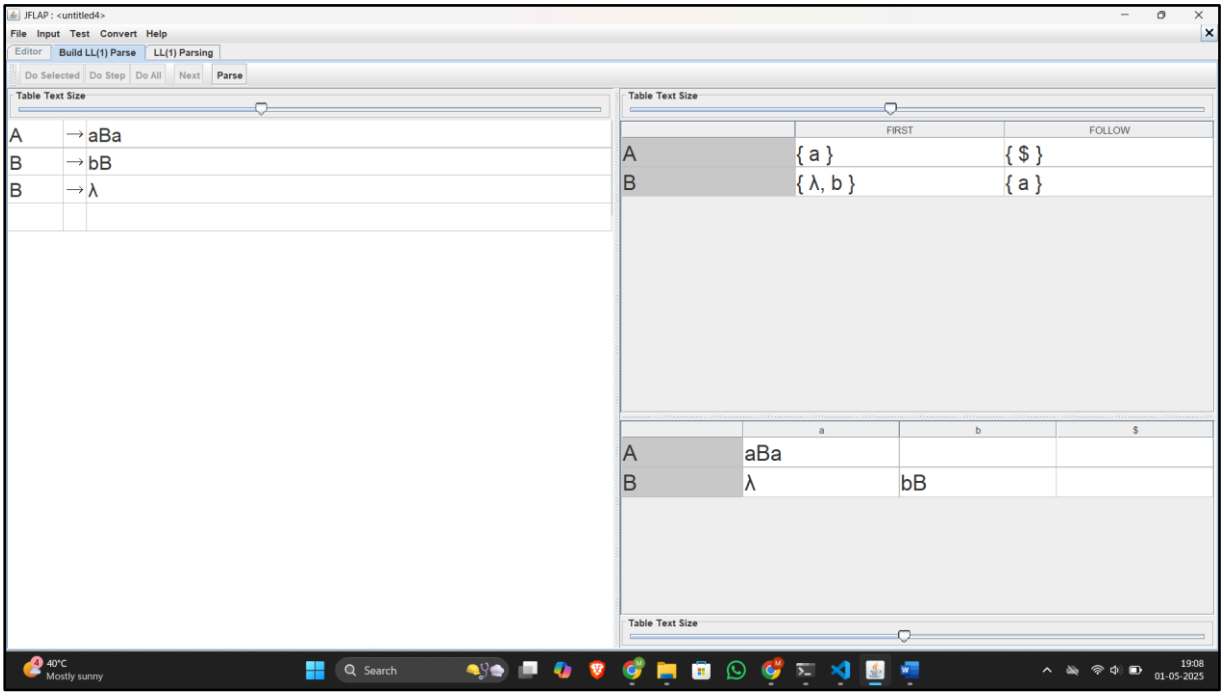
When a production can derive ε, entries in the parsing table are made using the FOLLOW set of that non-terminal. This allows the parser to know when it is valid to skip a non-terminal.

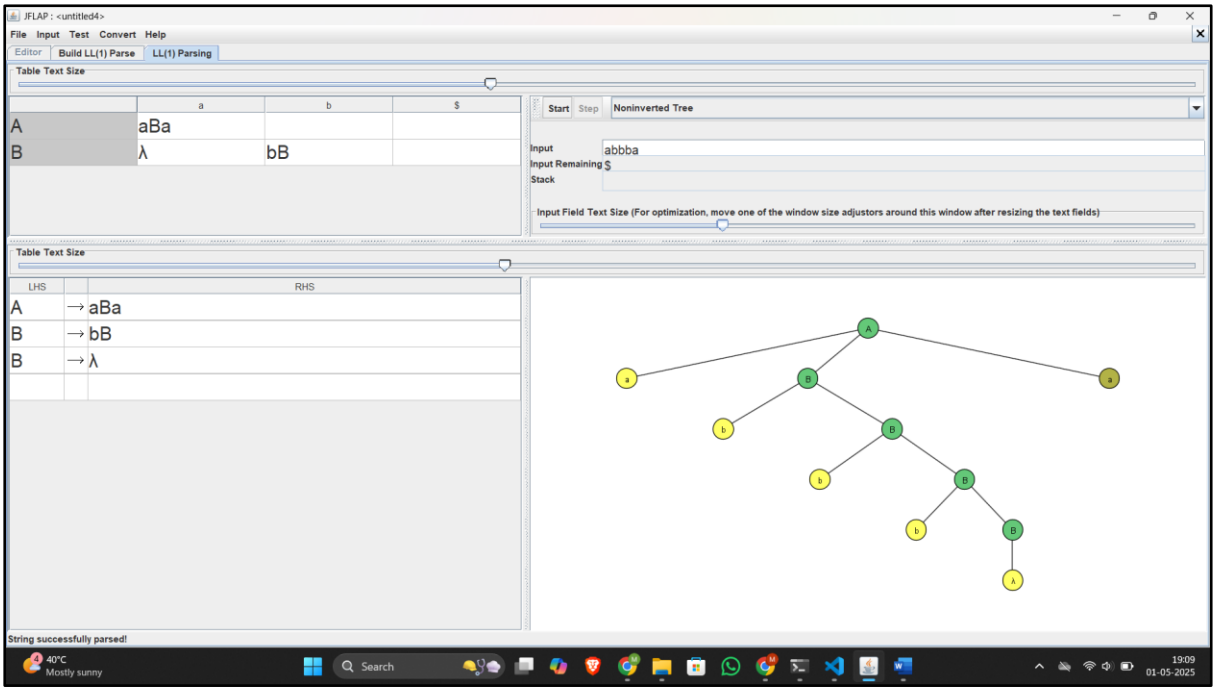
JFLAP Simulation:

JFLAP is a software tool that allows the simulation of automata and grammars. While JFLAP does not provide a direct option to create LL(1) parsing tables, it allows us to simulate **top-down parsing** by defining grammar rules and testing whether a given string can be parsed using those rules.

- In JFLAP, we can input a grammar and simulate the parsing process step by step.
- By selecting **"Top Down Derivation"**, we can simulate how an LL(1) parser would apply production rules and derive a string.
- The **parse tree** generated by JFLAP visually shows the steps in the derivation, which is similar to how an LL(1) parser would construct the parse tree based on the parsing table.

Output:





Conclusion:

In this practical, we learned how to construct FIRST and FOLLOW sets, use them to build a predictive LL(1) parsing table, and simulate parsing using JFLAP. This helped us understand how top-down parsers work and how to design grammars suitable for LL(1) parsing.