Compiler Design LAB
Autumn Semester 2025-26
Assignment 9

Name – Soham Chakraborty
Roll No – 22CS02002

Ans - 1) To create the AST, I created a struct having attributes

- Value (stores value at the current node)
- Operator (if node is non-Terminal i.e. operator)
- Id (index of that node)
- Left (node id to the left of current node)
- Node (node id to the right of current node)

All attributes are synthesized

Semantics –

For each operation of type – **L op R**

We create a node with value as evaluated from the above expression.

Assign id := number of current nodes + 1

Left = L.id;   Right = R.id

- For unary minus we don’t create a new node and instead just add a ‘-‘ to the value
- For parentheses we also don’t create a new node and instead assign use the same node values to those on the right (production side).

Ans - 2) To create the DAG, it is the same as AST but instead I also maintain another attribute **postfix** which is the postfix expression upto that node.

This is used if two nodes have the same postfix expression then we replace the second node and instead make it point to the first node.

Also, we introduce delimiters **$** for maintaining postfix expression to avoid ambiguity.

The remaining procedure is the same as in AST including the semantics.