

**BANGLADESH UNIVERSITY OF BUSINESS AND TECHNOLOGY
(BUBT)**



Lab Report

Course Code : CSE 324
Course Title : Compiler Design Lab
Date of Submission : March 10, 2024

Submitted By

Name : Aktaruzzaman
ID : 21222203031
Intake : 41
Section : 1

Submitted To

Ms. Adeeba Anis

Lecturer

Department of Computer Science &
Engineering

Bangladesh University of Business and
Technology (BUBT)

Experiment No: 06

Experiment Name: Finding First of any Grammar.

Problem Structure

The task is to design and implement a C/C++ program to find the First set of any grammar. The First set of a non-terminal in a grammar contains all terminals that can appear as the first symbol of any string derived from that non-terminal.

Procedure

- Define the grammar by specifying the productions for each non-terminal.
- Implement a function to compute the First set for each non-terminal recursively.
- Traverse each production of a non-terminal:
 - If the first symbol is a terminal, add it to the First set.
 - If the first symbol is a non-terminal:
 - Recursively compute the First set for that non-terminal.
 - Add the First set of that non-terminal to the First set of the current non-terminal.
 - If the non-terminal is nullable, continue to the next symbol in the production.
 - If the non-terminal is not nullable, stop and proceed to the next production.
- Repeat step 3 for all productions of each non-terminal.
- Output the First set for each non-terminal.

Code:

```
FindFirst.cpp x
1  #include <iostream>
2  #include <vector>
3  #include <unordered_set>
4  #include <unordered_map>
5
6  using namespace std;
7
8  unordered_map<char, vector<string>> productions;
9  unordered_set<char> nonTerminals;
10 unordered_set<char> nullable;
11
12 unordered_set<char> first(char nonTerminal) {
13     unordered_set<char> firstSet;
14
15     if (!nonTerminals.count(nonTerminal)) {
16         firstSet.insert(nonTerminal);
17         return firstSet;
18     }
19
20     for (auto production : productions[nonTerminal]) {
21         bool flag = true;
22         for (char symbol : production) {
23             if (nonTerminals.count(symbol)) {
```

```

24         auto firstSymbol = first(symbol);
25         firstSet.insert(firstSymbol.begin(), firstSymbol.end());
26         if (!nullable.count(symbol)) {
27             flag = false;
28             break;
29         }
30     } else {
31         firstSet.insert(symbol);
32         flag = false;
33         break;
34     }
35 }
36 if (flag) {
37     firstSet.insert('@');
38 }
39 }
40 return firstSet;
41 }
42
43 int main() {
44     productions['E'] = {"TX"};
45     productions['X'] = {"+TX", "@"};
46     productions['T'] = {"FY"};
47     productions['Y'] = {"*FY", "@"};
48     productions['F'] = {"(E)", "i"};
49
50     nonTerminals = {'E', 'X', 'T', 'Y', 'F'};
51
52
53     nullable = {'X', 'Y'};
54
55     for (char nonTerminal : nonTerminals) {
56         cout << "First(" << nonTerminal << ") = { ";
57         auto firstSet = first(nonTerminal);
58         for (char symbol : firstSet) {
59             cout << symbol << " ";
60         }
61         cout << ")\n";
62     }
63
64     return 0;
65 }
66
67

```

Output

"C:\Users\Aktaruzzaman\Documents\6th Semester\4.1_1"

```

First(Y) = { @ * }
First(F) = { i ( }
First(T) = { ( i }
First(X) = { @ + }
First(E) = { i ( }

```