**SSN COLLEGE OF ENGINEERING, KALAVAKKAM**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**UCS1602 - Compiler Design Programming**

**Assignment-3**

**Implementation of Left Recursion Elimination**

**Name:** Rahul Ram M

**Class:** CSE - B

**Reg No:** 185001121

**Date:** 23/02/2021

**CODE:**

**ex\_03.main**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <stdbool.h>

char\* getSubString(char line[], int start, int stop)

{

int length = stop - start + 1;

char \*sub = (char\*)malloc(sizeof(char) \* (length + 1));

int c = 0;

while (c < length) {

sub[c] = line[start+c];

c++;

}

sub[c] = '\0';

return sub;

}

void eliminateLeftRecursion(char eq[])

{

char NT = eq[0];

char productions[10][15];

int num = 0, start = 2;

// extracting right productions and placing it in string array

for(int i = 2; i < strlen(eq); i++)

{

if(eq[i] == '|')

{

strcpy(productions[num++], getSubString(eq, start, i-1));

start = i + 1;

}

}

if(start < strlen(eq))

{

strcpy(productions[num++], getSubString(eq, start, strlen(eq)-1));

}

// checking for left recursion

bool isleftrecursive = false;

for(int i = 0; i < num; i++)

{

if(NT == productions[i][0])

{

isleftrecursive = true;

break;

}

}

if(!isleftrecursive)

{

printf("%s\n", eq);

return;

}

// eliminating left recursion

bool nonterminal = false, isFirst = true;

printf("%c=", NT);

for(int i = 0; i < num; i++)

{

if(NT != productions[i][0])

{

if(!isFirst)

{

printf("|");

}

else

{

isFirst = false;

}

printf("%s%c\'", productions[i], NT);

nonterminal = true;

}

}

if(!nonterminal)

{

printf("%c'\n", NT);

}

else

{

printf("\n");

}

printf("%c'=", NT);

isFirst = true;

for(int i = 0; i < num; i++)

{

if(NT == productions[i][0])

{

if(!isFirst)

{

printf("|");

}

else

{

isFirst = false;

}

printf("%s%c\'", getSubString(productions[i], 1, strlen(productions[i])-1), NT);

}

}

printf("|e\n");

}

int main()

{

FILE \*file = fopen ("input.txt", "r");

char eq[30];

while (fgets(eq, 30, file))

{

eq[strcspn(eq, "\n")] = 0;

eliminateLeftRecursion(eq);

}

}

**Sample Input:**

E=E+T|T

T=T\*F|F

F=id|(E)

**Sample Output:**

E=TE'

E'=+TE'|e

T=FT'

T'=\*FT'|e

F=id|(E)

**Learning Outcomes:**

This assignment helped me

1. To understand the need for elimination of left recursion.

2. To understand the rules in the elimination of left recursion.

3. To implement the elimination of left recursion.