

LEFT FACTOR AND LEFT RECURSION ELIMINATION

M RAJASOUMYA

ch.en.u4cse22049

Experiment No: 02

Aim: To implement eliminate left recursion and left factoring from the given grammar using C program

CODE

```
UW PICO 5.09

#include <stdio.h>
#include <string.h>

void eliminateLeftRecursion(char A, char alpha[], char beta[]) {
    printf("\nA → A%s | %s\n", alpha, beta);
    printf("= %c → %s%c'\n", A, beta, A);
    printf("  %c' → %s%c' | ε\n", A, alpha, A);
}

void leftFactoring(char A, char p1[], char p2[]) {
    int i=0; char pre[20];
    while(p1[i]&&p2[i]&&p1[i]==p2[i]){ pre[i]=p1[i]; i++; }
    pre[i]='\0';
    printf("\n%c → %s | %s\n", A, p1, p2);
    printf("= %c → %s%c'\n", A, pre, A);
    printf("  %c' → %s | %s\n", A, p1+i, p2+i);
}

int main() {
    eliminateLeftRecursion('A',"x","y");
    leftFactoring('B',"abcX","abcY");
}
```

OUTPUT

```
[(base) rajasoumya@Rajasoumyas-MacBook-Air ~ % gcc lrr.c
[(base) rajasoumya@Rajasoumyas-MacBook-Air ~ % ./a.out

A → Ax | y
= A → yA'
  A' → xA' | ε

B → abcX | abcY
= B → abcB'
  B' → X | Y

[(base) rajasoumya@Rajasoumyas-MacBook-Air ~ %
[(base) rajasoumya@Rajasoumyas-MacBook-Air ~ %
```

RESULT

The left recursion and left factoring is eliminated for the given grammar using c program.