

Programming Assignment-3

Implementation of left recursion elimination

Gayathri M
185001050

Input file contents(input.txt):

E->E+T|T
T->T*F|F
F->(E)|id

Program Code:

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<ctype.h>

int main()
{
    char ch, buffer[1000],b[100],str[10][5];
    FILE *fp;
    int i,j=0,s,l,arr[10];
    fp = fopen("input.txt","r");
    if(fp == NULL)
    {
        printf("error while opening the file\n");
        exit(0);
    }
    int k=0;
    while((ch = fgetc(fp)) != EOF)
    {
        buffer[k++]=ch;
    }
    buffer[k]='\0';
    k=0;
    int t=0;
    while(buffer[k]!='\0')
    {
        t=0;
        while(buffer[k]!='\n')
        {
            b[t++]=buffer[k++];
        }
    }
}
```

```

b[t]='\0';
printf("\n\nProduction: %s",b);
ch=b[0];
t=3;
for(l=3;b[l]!='\0';l++)
{
    b[l-3]=b[l];
}
b[l-3]='\0';
j=0;
char *token = strtok(b, "|");
s=0;
while( token != NULL )
{
    strcpy(str[s++],token);
    int len=0;
    while (*token != '\0') {
len++;
token++;
    }
    arr[s-1]=len;
    token = strtok(NULL, "|");
}
for(i=0;i<s;i++)
{
    if(str[i][0]==ch)
        j=1;
}
if(j==1)
{
    printf("\n%c->",ch);
    for(i=0;i<s;i++)
    {
        if(str[i][0]!=ch)
        {
            printf("%s%c\\",str[i],ch);
            if(s-1!=i)
                printf("|");
        }
    }
    printf("\n%c\\'->",ch);
    for(i=0;i<s;i++)
    {
        if(str[i][0]==ch)
        {
            int h=1;
            for(h=1;h<arr[i];h++)
            {

```

```

                                printf("%c",str[i][h]);
                                }
                                printf("%c\\",ch);
                                }
                                }
                                printf("epsilon");
                                }
                                memset(b,0,strlen(b));
                                k++;
                                }

                                fclose(fp);
                                return 0;
                                }

```

/*SAMPLE INPUT/OUTPUT

Production: E->E+T|T
E->TE'
E'->+TE'|epsilon

Production: T->T*F|F
T->FT'
T'->*FT'|epsilon

Production: F->(E)|id

*/