Harshit Aggarwal (RA1911003010782)

Experiment 7: Shift Reduce Parsing

9/3/22

Aim:
To implement shift reduce parser using C program.

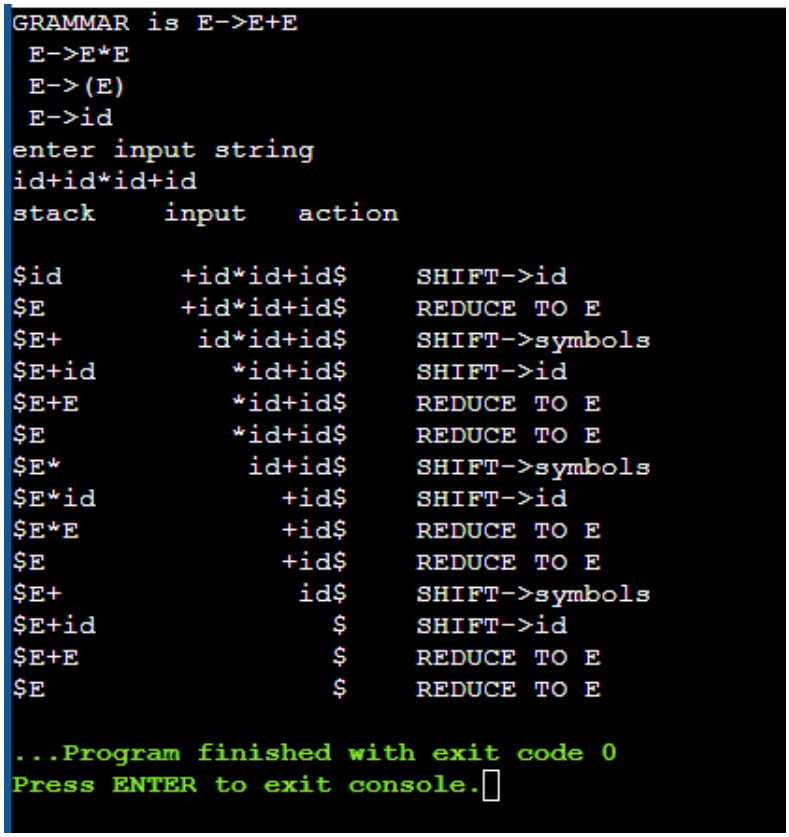
Algorithm:
1. Start
2. Get the input expression and store it in the input buffer.
3. Read the data from the input buffer one at the time
4. Using stack and push & pop operation shift and reduce symbols with respect to production rules available.
5. Continue the process till symbol shift and production rule reduce reaches the start symbol.
6. Display the Stack Implementation table with corresponding Stack actions with input symbols.
7. Stop

Code:
```
#include<stdio.h>
#include<string.h>
int k=0,z=0,i=0,j=0,c=0;
char a[16],ac[20],stk[15],act[10];
void check();
int main()
  {

    puts("GRAMMAR is E->E+E \n E->E*E \n E->(E) \n E->id");
    puts("enter input string ");
    scanf("%s",a);
    c=strlen(a);
    strcpy(act,"SHIFT->");
    puts("stack \t input \t action");
    for(k=0,i=0; j<c; k++,i++,j++)
     {
       if(a[j]=='i' && a[j+1]=='d')
        {
          stk[i]=a[j];
          stk[i+1]=a[j+1];
          stk[i+2]='\0';
          a[j]=' ';
          a[j+1]=' ';
```

```
            printf("\n$%s\t%s$\t%sid",stk,a,act);
            check();
          }
        else
          {
            stk[i]=a[j];
            stk[i+1]='\0';
            a[j]=' ';
            printf("\n$%s\t%s$\t%ssymbols",stk,a,act);
            check();
          }
      }

  }
void check()
  {
    strcpy(ac,"REDUCE TO E");
    for(z=0; z<c; z++)
      if(stk[z]=='i' && stk[z+1]=='d')
        {
          stk[z]='E';
          stk[z+1]='\0';
          printf("\n$%s\t%s$\t%s",stk,a,ac);
          j++;
        }
    for(z=0; z<c; z++)
      if(stk[z]=='E' && stk[z+1]=='+' && stk[z+2]=='E')
        {
          stk[z]='E';
          stk[z+1]='\0';
          stk[z+2]='\0';
          printf("\n$%s\t%s$\t%s",stk,a,ac);
          i=i-2;
        }
    for(z=0; z<c; z++)https://www.onlinegdb.com/#tab-stdin
      if(stk[z]=='E' && stk[z+1]=='*' && stk[z+2]=='E')
        {
          stk[z]='E';
          stk[z+1]='\0';
          stk[z+1]='\0';
          printf("\n$%s\t%s$\t%s",stk,a,ac);
          i=i-2;
        }
    for(z=0; z<c; z++)
```

```
    if(stk[z]=='(' && stk[z+1]=='E' && stk[z+2]==')')
     {
       stk[z]='E';
       stk[z+1]='\0';
       stk[z+1]='\0';
       printf("\n$%s\t%s$\t%s",stk,a,ac);
       i=i-2;
     }
 }
```

Output:



Result:

Hence Shift Reduce Parsing is doneFor Given Grammar.