Experiment 1

18CSC304J - Compiler Design

LEXICAL ANALYSER

Rahul Goel

RA1911030010094

Section: O2
Date: 14/01/2022

AIM: To study and code a lexical analyser in any of the programming languages.

Language used: c++

Software used: online c++ compiler

Algorithm:

• Start.
• Get the input program from the file prog.txt.
•
Readtheprogramlinebylineandcheckifeachwordinaline isa

keyword, identifier, constant or an operator.
•
Ifthewordreadisanidentifier,assignanumbertotheidentifierand

make an entry into the symbol table stored in sybol.txt. • For each lexeme read, generate a token as follows:

- If the lexeme is an identifier, then the token generated is of the form
- If the lexeme is an operator, then the token generated is .
- If the lexeme is a constant, then the token generated is .
- If the lexeme is a keyword, then the token is the keyword itself.

• The stream of tokens generated are displayed in the console output. • Stop.

Code: Main.cpp:

```
#include<bits/stdc++.h> #include<stdlib.h> #include<string.h>
#include<ctype.h>

using namespace std;

int isKeyword(char buffer[]){
char keywords[32][10] =
{"auto","break","case","char","const","continue","default",

"do","double","else","enum","extern","float","for","goto",
"if","int","long","register","return","short","signed",
"sizeof","static","struct","switch","typedef","union",
"unsigned","void","volatile","while"};

int i, flag = 0;

for(i = 0; i < 32; ++i){ if(strcmp(keywords[i], buffer) == 0){

flag = 1;
```

```
break; }

}

return flag; }

int main(){
char ch, buffer[15],b[30], logical_op[] = "><",math_op[]="+-*/
=",numer[]=".0123456789",other[]=",;\(){}[]":"; int count=0;
ifstream fin("Program.txt");
int mark[1000]={0};
int i,j=0,kc=0,ic=0,lc=0,mc=0,nc=0,oc=0,aaa=0;
vector < string > k;
vector<char >id;
vector<char>lo;
vector<char>ma;
vector<string>nu;

vector<char>ot; if(!fin.is_open()){

cout<<"error while opening the file\n";

exit(0); }

while(!fin.eof()){ ch = fin.get();

for(i = 0; i < 12; ++i){ if(ch == other[i]){

int aa=ch; if(mark[aa]!=1){

ot.push_back(ch); mark[aa]=1; ++oc;

} }

}

for(i = 0; i < 5; ++i){
if(ch == math_op[i]){

int aa=ch; if(mark[aa]!=1){

ma.push_back(ch); mark[aa]=1;
++mc;

} }

}
for(i = 0; i < 2; ++i){

if(ch == logical_op[i]){ int aa=ch;
```

```
if(mark[aa]!=1){ lo.push_back(ch); mark[aa]=1;
++lc;

} }

}

if(ch=='0' || ch=='1' || ch=='2' || ch=='3' || ch=='4' || ch=='5' || ch=='6' ||
ch=='7' || ch=='8' || ch=='9' || ch=='.' ||ch == ' ' || ch == '\n' || ch == ';'){

if(ch=='0' || ch=='1' || ch=='2' || ch=='3' || ch=='4' || ch=='5' || ch=='6' ||
ch=='7' || ch=='8' || ch=='9' || ch=='.')b[aaa++]=ch;

if((ch == ' ' || ch == '\n' || ch == ';') && (aaa != 0)){ b[aaa] = '\0';

aaa = 0; char arr[30];

strcpy(arr,b); nu.push_back(arr);

++nc;

} }

if(isalnum(ch)){ buffer[j++] = ch;

}
else if((ch == ' ' || ch == '\n') && (j != 0)){

buffer[j] = '\0'; j = 0;

if(isKeyword(buffer) == 1){

k.push_back(buffer);

++kc; }

else{

if(buffer[0]>=97 && buffer[0]<=122) { if(mark[buffer[0]-'a']!=1)
{ id.push_back(buffer[0]);
++ic;

mark[buffer[0]-'a']=1; }

} }

} }

fin.close(); printf("Keywords: ");

for(int f=0;f<kc;++f){ if(f==kc-1){
```

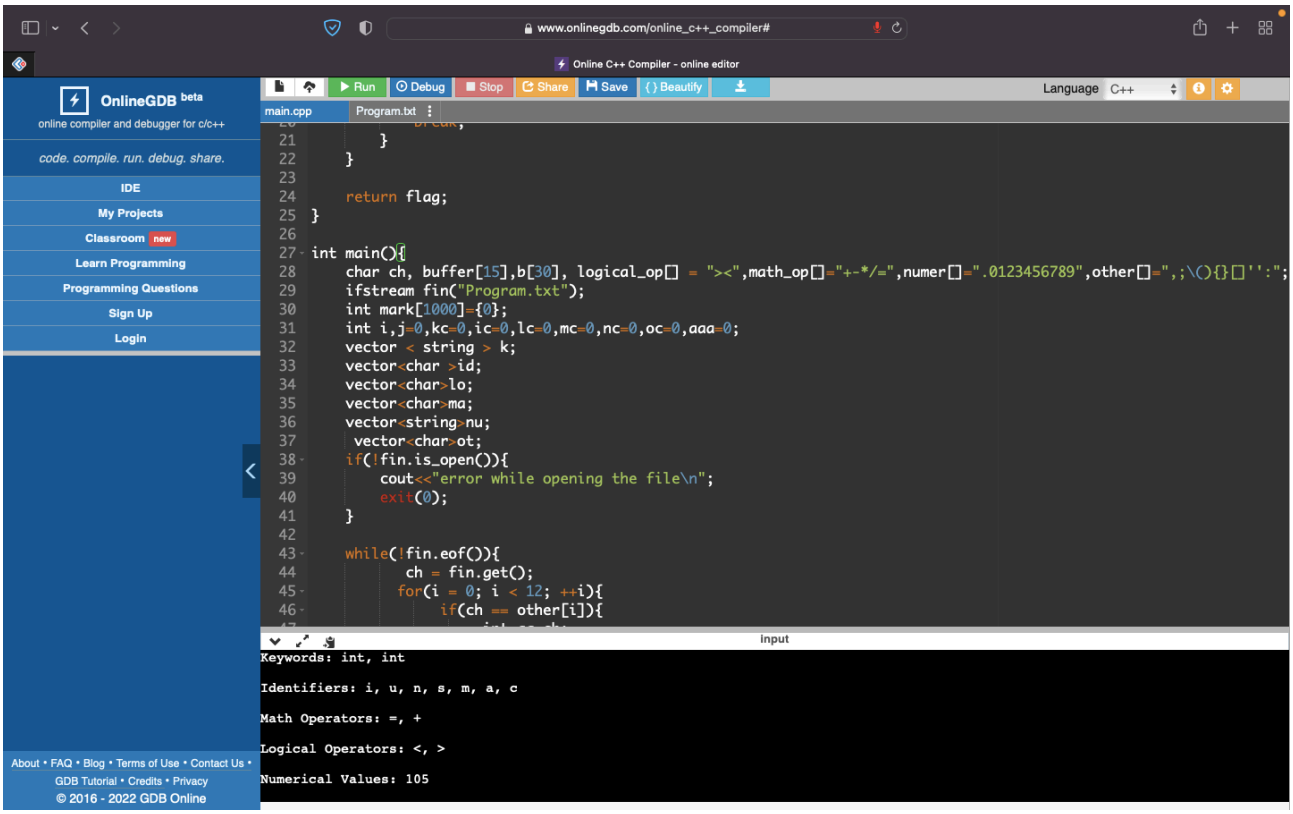
```
cout<<k[f]<<"\n";

count++; }

else { cout<<k[f]<<", ";

} }

cout<<"no.of Keywords "<<count+1<<endl; count=0;
printf("\nIdentifiers: ");

for(int f=0;f<ic;++f){ if(f==ic-1){

cout<<id[f]<<"\n";

count++; }

else { cout<<id[f]<<", ";

} }

cout<<"no.of Identifiers: "<<count+1<<endl; count=0;
printf("\nMath Operators: ");
for(int f=0;f<mc;++f){

if(f==mc-1){ cout<<ma[f]<<"\n"; count++;

}
else {

cout<<ma[f]<<", "; }

}
cout<<"no.of Math Operators: "<<count+1<<endl; count=0;
printf("\nLogical Operators: ");
for(int f=0;f<lc;++f){

if(f==lc-1){ cout<<lo[f]<<"\n"; count++;

}
else {

cout<<lo[f]<<", "; }

}
cout<<"no.of Logical Operators: "<<count+1<<endl;

count=0;
printf("\nNumerical Values: "); for(int f=0;f<nc;++f){
```

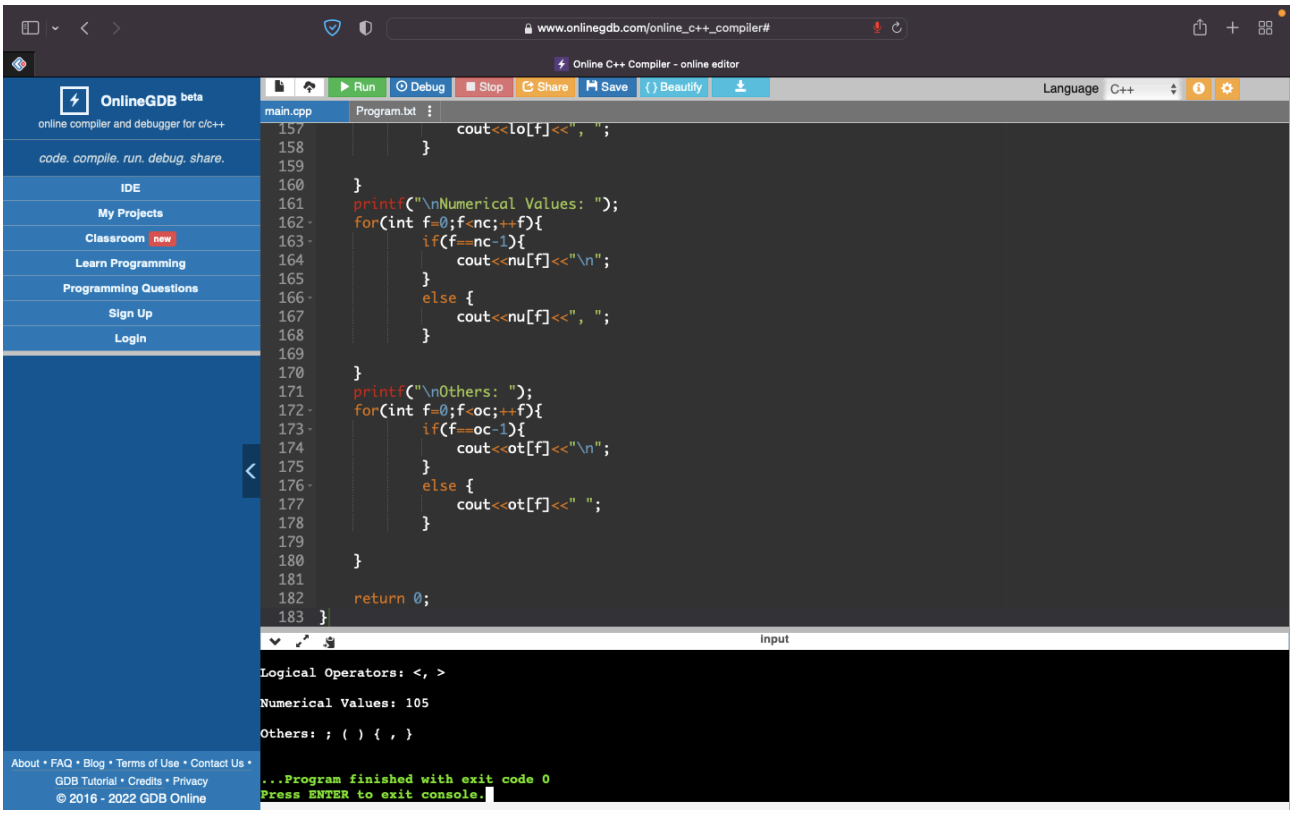
```
if(f==nc-1){
cout<<nu[f]<<"\n";
count++; }
else { cout<<nu[f]<<", ";
}
}
cout<<"no.of Numerical Values: "<<count+1<<endl;
count=0; printf("\nOthers: "); for(int f=0;f<oc;++f){
if(f==oc-1){ cout<<ot[f]<<"\n"; count++;
}
else {
cout<<ot[f]<<" "; }
}
cout<<"no.of Others: "<<count+1<<endl;
return 0; }
```

Program.txt:

```
#include<iostream> int main()
{
int a=5,b=10; cout<<a+b;
}
```

Output:


www.onlinegdb.com/online_c++_compiler#
Online C++ Compiler - online editor
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
Run
Debug
Stop
Share
Save
{} Beautify
Language C++
main.cpp
Program.txt
21 }
22 }
23
24 return flag;
25 }
26
27 int main(){
28 char ch, buffer[15],b[30], logical_op[] = "><",math_op[]="+-*/=",numer[]=".0123456789",other[]=",;\(){}[]'':";
29 ifstream fin("Program.txt");
30 int mark[1000]={0};
31 int i,j=0,kc=0,ic=0,lc=0,mc=0,nc=0,oc=0,aaa=0;
32 vector < string > k;
33 vector<char >id;
34 vector<char>lo;
35 vector<char>ma;
36 vector<string>nu;
37 vector<char>ot;
38 if(!fin.is_open()){
39 cout<<"error while opening the file\n";
40 exit(0);
41 }
42
43 while(!fin.eof()){
44 ch = fin.get();
45 for(i = 0; i < 12; ++i){
46 if(ch == other[i]){
input
Keywords: int, int
Identifiers: i, u, n, s, m, a, c
Math Operators: =, +
Logical Operators: <, >
Numerical Values: 105
About • FAQ • Blog • Terms of Use • Contact Us •
GDB Tutorial • Credits • Privacy
© 2016 - 2022 GDB Online


www.onlinegdb.com/online_c++_compiler#
Online C++ Compiler - online editor
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
Run
Debug
Stop
Share
Save
{} Beautify
Language C++
main.cpp
Program.txt
157 cout<<lo[f]<<", ";
158 }
159
160 }
161 printf("\nNumerical Values: ");
162 for(int f=0;f<nc;++f){
163 if(f==nc-1){
164 cout<<nu[f]<<"\n";
165 }
166 else {
167 cout<<nu[f]<<", ";
168 }
169
170 }
171 printf("\nOthers: ");
172 for(int f=0;f<oc;++f){
173 if(f==oc-1){
174 cout<<ot[f]<<"\n";
175 }
176 else {
177 cout<<ot[f]<<" ";
178 }
179
180 }
181
182 return 0;
183 }
input
Logical Operators: <, >
Numerical Values: 105
Others: ; ( ) { , }
...Program finished with exit code 0
Press ENTER to exit console.
About • FAQ • Blog • Terms of Use • Contact Us •
GDB Tutorial • Credits • Privacy
© 2016 - 2022 GDB Online

Keywords: int, int no.of Keywords 2

Identifiers: i, m, a, c no.of Identifiers: 2

Math Operators: =, + no.of Math Operators: 2

Logical Operators: <, > no.of Logical Operators: 2

Numerical Values: 510 no.of Numerical Values: 2

Others: ( ) { , ; } no.of Others: 2

Result: Lexical Analyser was studied and executed successfully in c++.

Experiment 2

18CSC304J - Compiler Design

Regular Expression to NFA conversion

Rahul Goel

RA1911030010094

Section: O2
Date: 4/02/2022

AIM: To study and perform regular expression to NFA(non deterministic automata) conversion in any of the programming languages.

Language used: c++
Software used: online c++ compiler

Algorithm:

- Start

- Get the input from the user

- Initialise separate variables and functions for Postfix, Display and
  NFA.

- Create separate methods for different operators like +,*,.

- By using Switch case initialise different cases for the input

- For '.' operator initialise a separate method by using various stack
  functions. Do the same for other operators like *,+

- Regular expression is in the form of a.b(or) a+b

- Display the output

- Stop

Code:

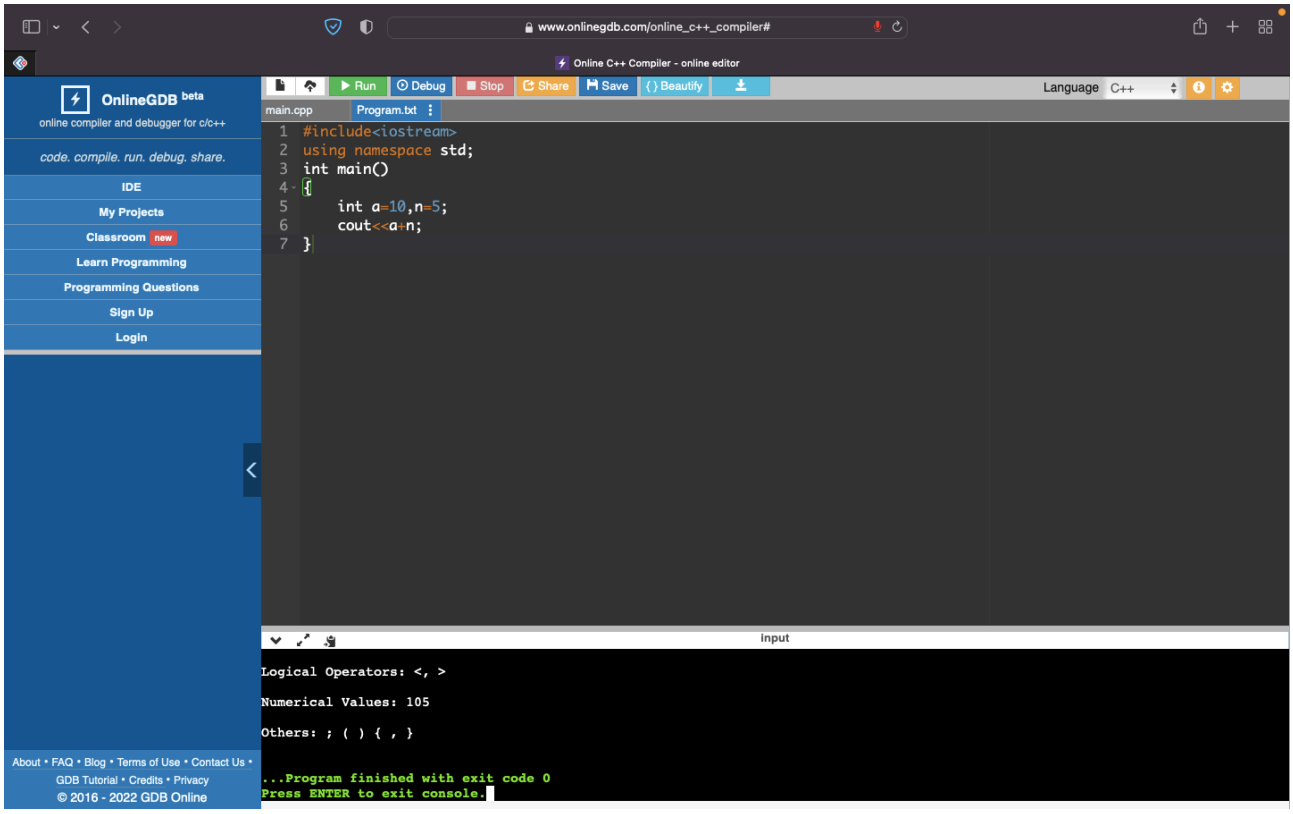
```
#include<iostream> #include<string.h> int main()
{
printf("Enter the regular expression: "); char reg[20];
int q[20][3],i,j,len,a,b; for(a=0;a<20;a++)
{
for(b=0;b<3;b++) {
q[a][b]=0; }
```

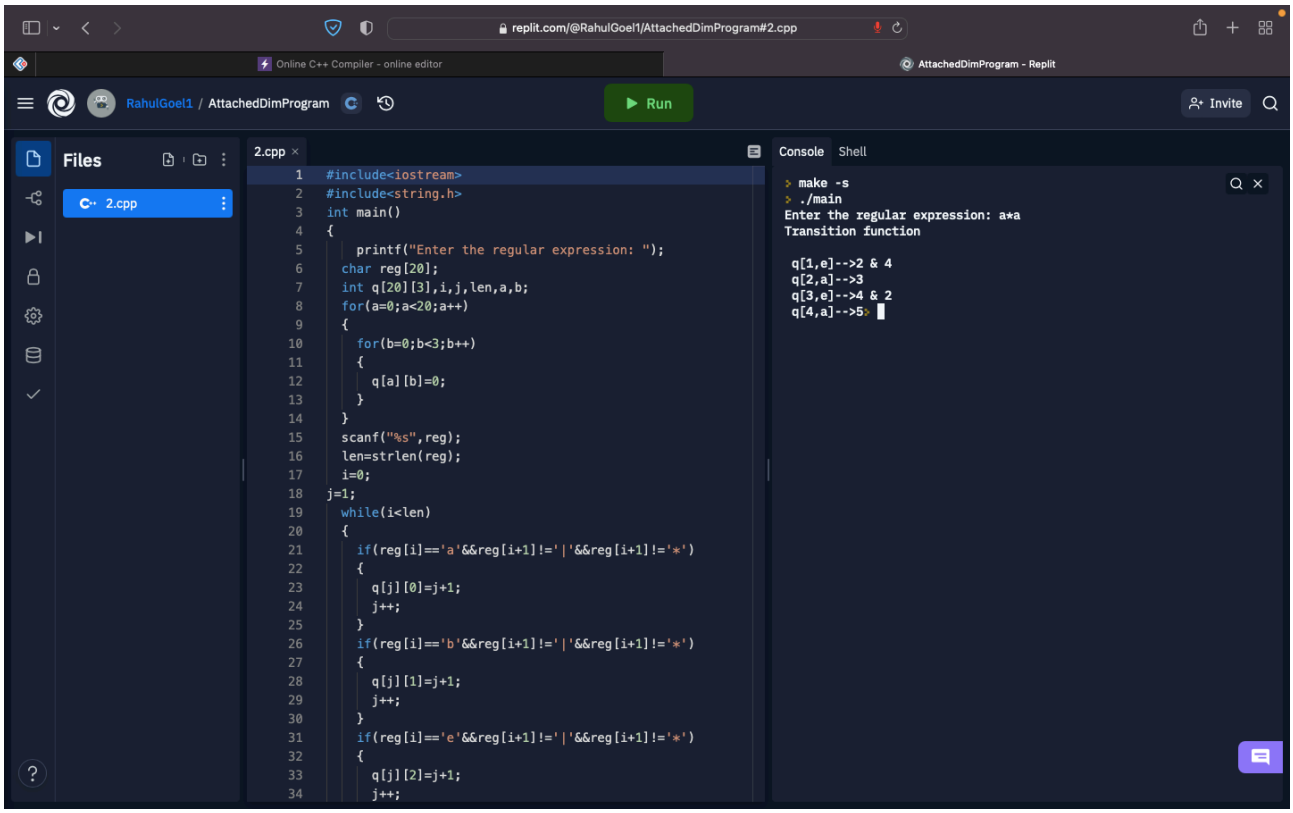
```
}
scanf("%s",reg); len=strlen(reg); i=0;
```

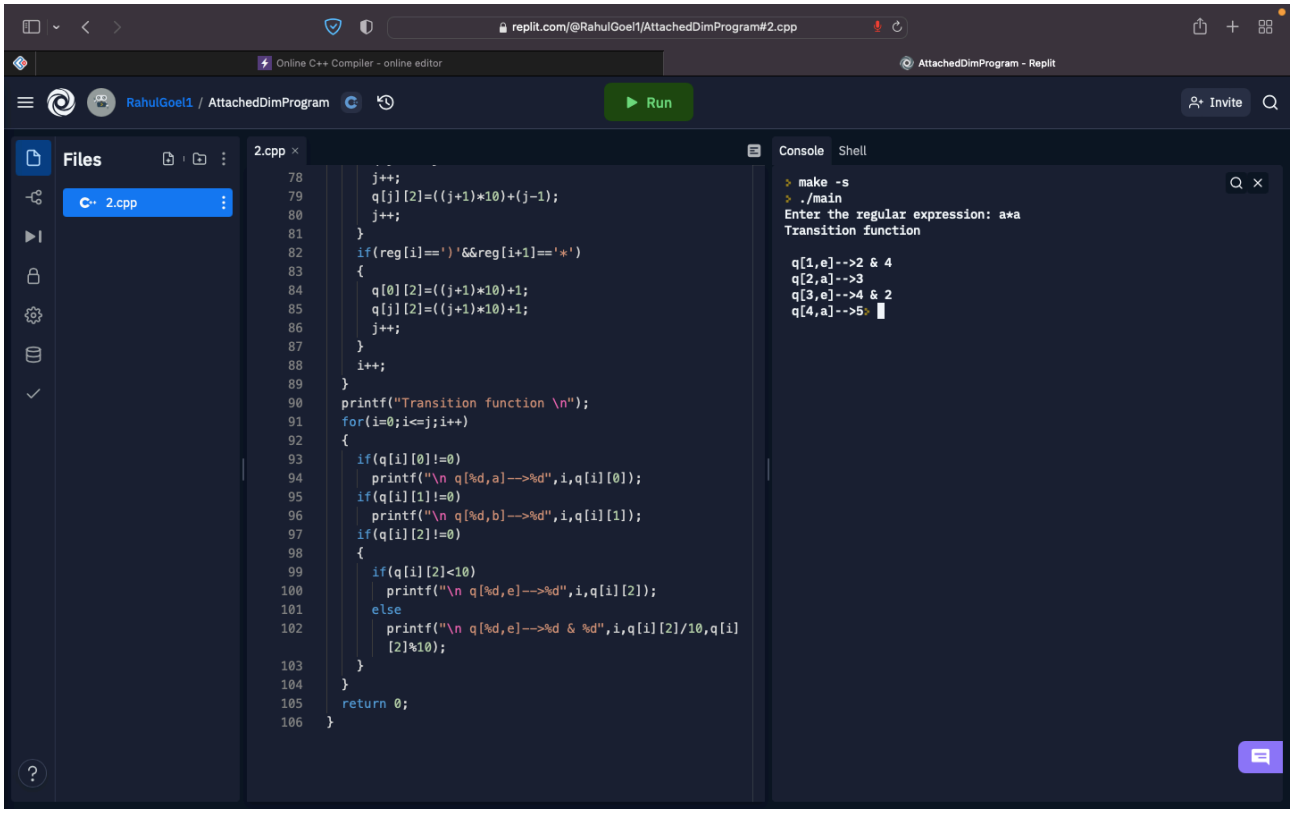
```
j=1;
while(i<len)

{
if(reg[i]=='a'&&reg[i+1]!='|'&&reg[i+1]!='*') {

q[j][0]=j+1;

j++; }

if(reg[i]=='b'&&reg[i+1]!='|'&&reg[i+1]!='*') {

q[j][1]=j+1;

j++; }

if(reg[i]=='e'&&reg[i+1]!='|'&&reg[i+1]!='*') {

q[j][2]=j+1;

j++; }

if(reg[i]=='a'&&reg[i+1]=='|'&&reg[i+2]=='b') {

q[j][2]=((j+1)*10)+(j+3); j++;
q[j][0]=j+1;
j++;

q[j][2]=j+3; j++; q[j][1]=j+1; j++; q[j][2]=j+1; j++;

i=i+2; }

if(reg[i]=='b'&&reg[i+1]=='|'&&reg[i+2]=='a') {

q[j][2]=((j+1)*10)+(j+3); j++;
q[j][1]=j+1;
j++;

q[j][2]=j+3; j++; q[j][0]=j+1; j++; q[j][2]=j+1; j++;

i=i+2; }

if(reg[i]=='a'&&reg[i+1]=='*') {

q[j][2]=((j+1)*10)+(j+3); j++;
q[j][0]=j+1;
j++; q[j][2]=((j+1)*10)+(j-1); j++;

} if(reg[i]=='b'&&reg[i+1]=='*') {
```

```
q[j][2]=((j+1)*10)+(j+3); j++;

q[j][1]=j+1;
j++; q[j][2]=((j+1)*10)+(j-1); j++;

} if(reg[i]==')'&&reg[i+1]=='*') {

q[0][2]=((j+1)*10)+1; q[j][2]=((j+1)*10)+1; j++;

}

i++; }

printf("Transition function \n"); for(i=0;i<=j;i++)
{

if(q[i][0]!=0)
printf("\n q[%d,a]-->%d",i,q[i][0]);

if(q[i][1]!=0)
printf("\n q[%d,b]-->%d",i,q[i][1]);

if(q[i][2]!=0) {

} }

return 0; }

if(q[i][2]<10)
printf("\n q[%d,e]-->%d",i,q[i][2]);

else
printf("\n q[%d,e]-->%d & %d",i,q[i][2]/10,q[i][2]%10);
```

Output:


www.onlinegdb.com/online_c++_compiler#
Online C++ Compiler - online editor
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
Run
Debug
Stop
Share
Save
{ } Beautify
Language C++
main.cpp
Program.txt
#include<iostream>
using namespace std;
int main()
{
int a=10,n=5;
cout<<a+n;
}
input
Logical Operators: <, >
Numerical Values: 105
Others: ; ( ) { , }
...Program finished with exit code 0
Press ENTER to exit console.
About • FAQ • Blog • Terms of Use • Contact Us •
GDB Tutorial • Credits • Privacy
© 2016 - 2022 GDB Online


replit.com/@RahulGoel1/AttachedDimProgram#2.cpp
Online C++ Compiler - online editor
AttachedDimProgram - Replit
RahulGoel1 / AttachedDimProgram
Run
Invite
Files
2.cpp
#include<iostream>
#include<string.h>
int main()
{
printf("Enter the regular expression: ");
char reg[20];
int q[20][3],i,j,len,a,b;
for(a=0;a<20;a++)
{
for(b=0;b<3;b++)
{
q[a][b]=0;
}
}
scanf("%s",reg);
len=strlen(reg);
i=0;
j=1;
while(i<len)
{
if(reg[i]=='a'&&reg[i+1]!='|'&&reg[i+1]!='*')
{
q[j][0]=j+1;
j++;
}
if(reg[i]=='b'&&reg[i+1]!='|'&&reg[i+1]!='*')
{
q[j][1]=j+1;
j++;
}
if(reg[i]=='e'&&reg[i+1]!='|'&&reg[i+1]!='*')
{
q[j][2]=j+1;
j++;
Console
Shell
> make -s
> ./main
Enter the regular expression: a*a
Transition function
q[1,e]-->2 & 4
q[2,a]-->3
q[3,e]-->4 & 2
q[4,a]-->5>



Enter the regular expression: (a/b)*b Transition function

q[0,e]-->4 & 1 q[1,a]-->2 q[2,b]-->3 q[3,e]-->4 & 1 q[4,b]-->5

Result:

The regular expression to NFA conversion was successfully executed in C++.

Experiment 3

18CSC304J - Compiler Design

NFA to DFA conversion

Rahul Goel

RA1911030010094

Section: O2
Date: 11/02/2022

AIM: To study and perform NFA(non deterministic automata) to DFA(deterministic automata) conversion in any of the programming languages.

Language used: c++

Software used: online c++ compiler

Algorithm:

1. In the main function we initialise a 3d-array vector to dynamically store the data for the transition table of NFA and then eventually convert it to DFA STEP

2. We take input for the total number of states in NFA and also total number of elements in the alphabet (no. of input symbols)

3. Then for each state we build nested for loops wherein for each state and for each input we take the number of output states STEP

4. Next whatever we get as a 3D array we insert it to the main 3D vector table STEP

5. We now call the print function wherein we display the 3d table elements

6. Next for the transition table of DFA we need to do the epsilon closure for that we need to compute 4
   nested for loops within a while loop

7. Use the graph adjacency technique to track if for any of the state whether it is directing to a new
   state/adjacency location(this indicates state change according to given input symbol).If yes we take
   that row data and store it into another 3-d matrix which contains the solution

8. Now print the dfa table by creating a printdfa function which displays the 3D array data

---

Code:

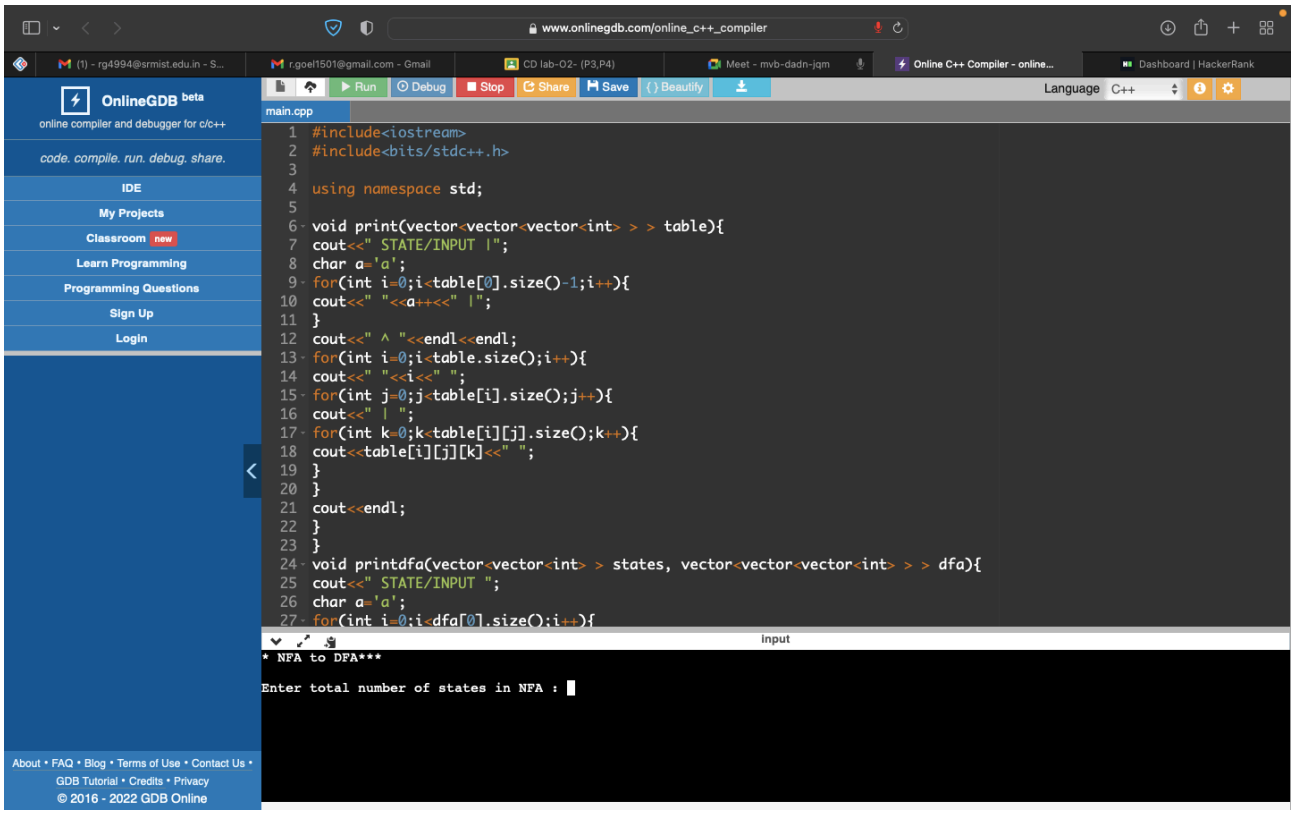
```
#include<iostream>
#include<bits/stdc++.h>
using namespace std;
void print(vector<vector<vector<int> > > table){ cout<<" STATE/INPUT |";

char a='a';
for(int i=0;i<table[0].size()-1;i++){
cout<<" "<<a++<<" |";
}
cout<<" ^ "<<endl<<endl;
for(int i=0;i<table.size();i++){
cout<<" "<<i<<" ";
for(int j=0;j<table[i].size();j++){
cout<<" | ";
for(int k=0;k<table[i][j].size();k++){
cout<<table[i][j][k]<<" ";
}
}
cout<<endl;
}
```

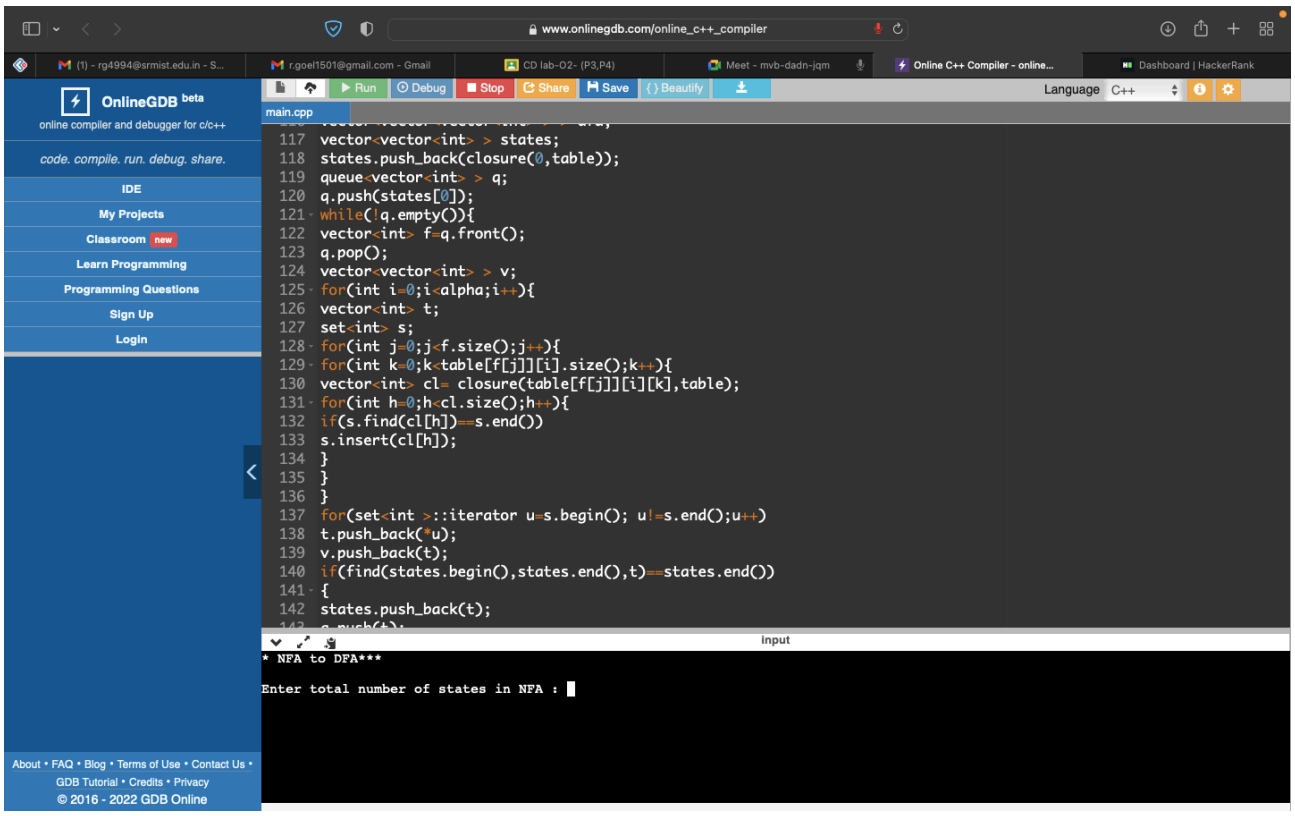
```
}
void printdfa(vector<vector<int> > states, vector<vector<vector<int> > >
dfa){ cout<<" STATE/INPUT ";
char a='a';
for(int i=0;i<dfa[0].size();i++){
cout<<"| "<<a++<<" ";
}
cout<<endl;
for(int i=0;i<states.size();i++){
cout<<"{ ";
for(int h=0;h<states[i].size();h++)
cout<<states[i][h]<<" ";
if(states[i].empty()){
cout<<"^ ";
}
cout<<"} ";
for(int j=0;j<dfa[i].size();j++){
cout<<" | ";
for(int k=0;k<dfa[i][j].size();k++){
cout<<dfa[i][j][k]<<" ";
}
if(dfa[i][j].empty()){
cout<<"^ ";
}
}
cout<<endl;
}
}
vector<int> closure(int s,vector<vector<vector<int> > > v){
vector<int> t;
queue<int> q;
t.push_back(s);
int a=v[s][v[s].size()-1].size();
for(int i=0;i<a;i++){
t.push_back(v[s][v[s].size()-1][i]);
//cout<<"t[i]"<<t[i]<<endl;
q.push(t[i]);
}

while(!q.empty()){
int f=q.front();
q.pop(); if(!v[f][v[f].size()-1].empty()){ int u=v[f][v[f].size()-1].size(); for(int
i=0;i<u;i++){

int y=v[f][v[f].size()-1][i]; if(find(t.begin(),t.end(),y)==t.end()){ //
cout<<"y"<<y<<endl; t.push_back(y);
```

```
q.push(y); }
}
}

}
return t;
}
int main(){
int n,alpha;
cout<<"********* NFA to DFA*********"<<endl<<endl;
cout<<"Enter total number of states in NFA : ";
cin>>n;
cout<<"Enter number of elements in alphabet(no of input symbols) : ";
cin>>alpha;
vector<vector<vector<int> > > table;
for(int i=0;i<n;i++){
cout<<"For state "<<i<<endl;
vector< vector< int > > v;
char a='a';
int y,yn;
for(int j=0;j<alpha;j++){
vector<int> t;
cout<<"Enter no. of output states for input "<<a++<<" : ";
cin>>yn;
cout<<"Enter output states :"<<endl;
for(int k=0;k<yn;k++){
cin>>y;
t.push_back(y);
}
v.push_back(t);
}
vector<int> t;
cout<<"Enter no. of output states for input ^ : ";
cin>>yn;
cout<<"Enter output states :"<<endl;
for(int k=0;k<yn;k++){
cin>>y;
t.push_back(y);
}
v.push_back(t);
table.push_back(v);
}
cout<<"** TRANSITION TABLE OF NFA **"<<endl;
print(table);
cout<<endl<<"** TRANSITION TABLE OF DFA **"<<endl;
```

```
vector<vector<vector<int> > > dfa;
vector<vector<int> > states;
states.push_back(closure(0,table));
queue<vector<int> > q;
q.push(states[0]);
while(!q.empty()){

vector<int> f=q.front(); q.pop(); vector<vector<int> > v; for(int
i=0;i<alpha;i++){ vector<int> t;

set<int> s;
for(int j=0;j<f.size();j++){
for(int k=0;k<table[f[j]][i].size();k++){
vector<int> cl= closure(table[f[j]][i][k],table);
for(int h=0;h<cl.size();h++){ if(s.find(cl[h])==s.end())
s.insert(cl[h]);
}
}
}
for(set<int >::iterator u=s.begin(); u!=s.end();u++) t.push_back(*u);
v.push_back(t); if(find(states.begin(),states.end(),t)==states.end()) {
states.push_back(t);
q.push(t);
}
}
dfa.push_back(v);
}
printdfa(states,dfa);
}
```

Output:


www.onlinegdb.com/online_c++_compiler
(1) - rg4994@srmist.edu.in - S...
r.goel1501@gmail.com - Gmail
CD lab-02- (P3,P4)
Meet - mvb-dadn-jqm
Online C++ Compiler - online...
Dashboard | HackerRank
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
Run
Debug
Stop
Share
Save
{ } Beautify
Language C++
main.cpp
#include<iostream>
#include<bits/stdc++.h>
using namespace std;
void print(vector<vector<vector<int> > > table){
cout<<" STATE/INPUT |";
char a='a';
for(int i=0;i<table[0].size()-1;i++){
cout<<" "<<a++<<" |";
}
cout<<" ^ "<<endl<<endl;
for(int i=0;i<table.size();i++){
cout<<" "<<i<<" ";
for(int j=0;j<table[i].size();j++){
cout<<" | ";
for(int k=0;k<table[i][j].size();k++){
cout<<table[i][j][k]<<" ";
}
}
cout<<endl;
}
}
void printdfa(vector<vector<int> > states, vector<vector<vector<int> > > dfa){
cout<<" STATE/INPUT ";
char a='a';
for(int i=0;i<dfa[0].size();i++){
Input
* NFA to DFA***
Enter total number of states in NFA :
About • FAQ • Blog • Terms of Use • Contact Us •
GDB Tutorial • Credits • Privacy
© 2016 - 2022 GDB Online


www.onlinegdb.com/online_c++_compiler
(1) - rg4994@srmist.edu.in - S...
r.goel1501@gmail.com - Gmail
CD lab-02- (P3,P4)
Meet - mvb-dadn-jqm
Online C++ Compiler - online...
Dashboard | HackerRank
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
Run
Debug
Stop
Share
Save
{ } Beautify
Language C++
main.cpp
vector<vector<int> > states;
states.push_back(closure(0,table));
queue<vector<int> > q;
q.push(states[0]);
while(!q.empty()){
vector<int> f=q.front();
q.pop();
vector<vector<int> > v;
for(int i=0;i<alpha;i++){
vector<int> t;
set<int> s;
for(int j=0;j<f.size();j++){
for(int k=0;k<table[f[j]][i].size();k++){
vector<int> cl= closure(table[f[j]][i][k],table);
for(int h=0;h<cl.size();h++){
if(s.find(cl[h])==s.end())
s.insert(cl[h]);
}
}
}
for(set<int >::iterator u=s.begin(); u!=s.end();u++)
t.push_back(*u);
v.push_back(t);
if(find(states.begin(),states.end(),t)==states.end())
{
states.push_back(t);
Input
* NFA to DFA***
Enter total number of states in NFA :
About • FAQ • Blog • Terms of Use • Contact Us •
GDB Tutorial • Credits • Privacy
© 2016 - 2022 GDB Online

```
1
Enter no. of output states for input ^ : 0
Enter output states :
For state 1
Enter no. of output states for input a : 0
Enter output states :
Enter no. of output states for input b : 1
Enter output states :
2
Enter no. of output states for input ^ : 0
```

```
Enter output states :
1 2
Enter no. of output states for input b : 1
Enter output states :
2
Enter no. of output states for input ^ : 0
Enter output states :
* TRANSITION TABLE OF NFA *
 STATE/INPUT | a | b | ^
```

```
* TRANSITION TABLE OF DFA *
 STATE/INPUT | a | b
{ 0 }  | ^  | 1
{ ^ }  | ^  | ^
{ 1 }  | ^  | 2
{ 2 }  | 1 2  | 2
{ 1 2 }  | 1 2  | 2
```

```
 STATE/INPUT | a | b | ^

0  |  | 1  |
1  |  | 2  |
2  | 1 2  | 2  |

* TRANSITION TABLE OF DFA *
 STATE/INPUT | a | b
{ 0 }  | ^  | 1
```

Result:
NFA to DFA conversion was successfully executed in c++.

REGEX COMPLETION

Account id: rg4994
Total number of regex questions solved: 47/47