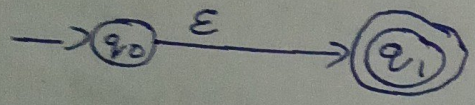
3. Finding ε-closure for NFA with ε-moves

Aim :-

To write a program to find ε-closure of a non-deterministic finite Automata with ε-move

Algorithm

1. Get the following Input from the user
   (i) number of states in NFA
   (ii) Number of Symbol Input alphabet Include ε.
   iii) Input Symbols
   iv) Number of final States and their names
2. Declare a 3-dimensional matrix to store the transitions and Initialize.
3. Get the transitions from every state for Input Symbol from the user and Store



4. Initialize of two-dimensional matrix ε-Closure with -1 in all entries
5. ε-closure of state q is defined of Set State that can be reached.

ε - closure (0) = 0, 1

ε - closure (1) = 1,

ε - closure (1) = 2,

6 for every state - point c- closure values

Program

```
#include <stdio.h>
#include<String.h>
  int trans-table [10][5][3];
  char Symbol[5], a;
  int e-closure [10][10], ptr, state;
  void find e - closure(int x);
  int main()
 {
  int i, j, k, n, num_state, num_ Symbols;
  for(i=0; i<10; i++)
  {
   for(j=0; j<5; j++)
   {
    for(k=0; k<3; k++)
    {
     trans_table[i][j][k] = -1;
    }
   }
 }
num_State = 3;
num - Symbols= 2;
```

```
Symbol[10] = 'e';
n=1;
trans_table[0][0][0]=-1;
for(i=0;i<10;i++)
{
for(j=0;j<10;j++)
{
e_closure[i][j]=-1;
}
}
for(i=0;i<num_state;i++)
e_closure[i][0]=i;
for(i=0;i<num_state;i++)
{
if(trans_table[i][0][0]==-1)
continue;
else
{
state=i;
ptr=1;
find_e_closure[i];
}
}
for(i=0;i<num_state;i++)
{
printf("e_closure(%d)=(%d,",i);
for(j=0;j<num_state;j++)
{
if(e_closure[i][j]!=-1)
{
```

```
Printf("%d", eclosure[i][j]);
}
}
Printf("\n");
}
}
void find_e-closure(int x)
{
int i,j,y[10], num_trans;
i=0;
while(trans_table[x][0][i]!=-1)
{
y[i] = trans_table[x][0][i];
i= i+1;
}
num_trans= 1;
for(j=0; j<num_trans; j++)
{
e_closure(state)[ptr]=y[j];
ptr++;
find_e_closure(y[j]);
}
```

Output

e_closure(0) = {0, 1}

e_closure(1) = {1}

e_closure(2) = {2}