**EXP 3 : CONVERSION OF NFA TO DFA**

**AIM:**

* To write a program for converting NFA to DFA using python language.

**ALGORITHM:**

* Start
* Get the input from the user
* Set the only state in SDFA to “unmarked”.
* while SDFA contains an unmarked state do:

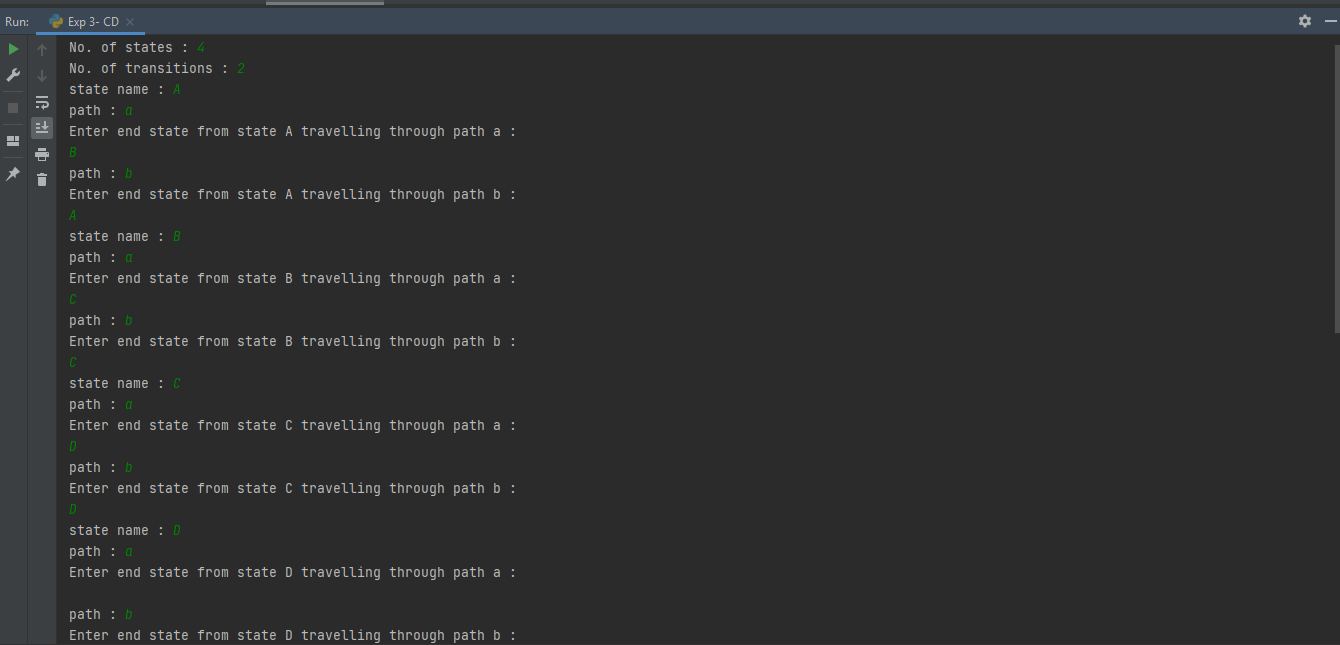
1. Let T be that unmarked state b.
2. for each a in % do S = e-Closure(MoveNFA(T,a))
3. if S is not in SDFA already then, add S to SDFA (as an “unmarked” state)
4. Set MoveDFA(T,a) to S

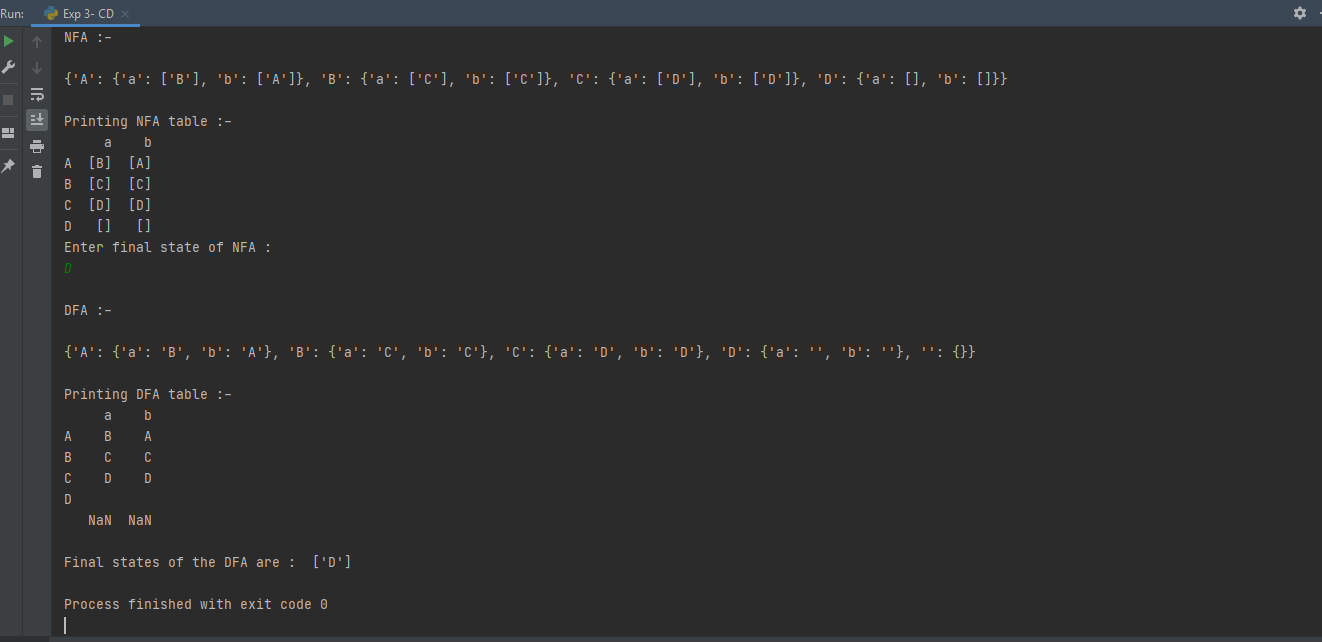
* For each S in SDFA if any s & S is a final state in the NFA then, mark S an a final state in the DFA
* Print the result.
* Stop the program

**SOURCE CODE :**

import pandas as pd  
  
nfa = {}  
n = int(input("No. of states : "))  
t = int(input("No. of transitions : "))  
for i in range(n):  
 state = input("state name : ")  
 nfa[state] = {}  
 for j in range(t):  
 path = input("path : ")  
 print("Enter end state from state {} travelling through path {} : ".format(state, path))  
 reaching\_state = [x for x in input().split()]  
 nfa[state][path] = reaching\_state  
print("\nNFA :- \n")  
print(nfa)  
print("\nPrinting NFA table :- ")  
nfa\_table = pd.DataFrame(nfa)  
print(nfa\_table.transpose())  
  
print("Enter final state of NFA : ")  
nfa\_final\_state = [x for x in input().split()]  
  
new\_states\_list = []  
dfa = {}  
keys\_list = list(list(nfa.keys())[0])  
path\_list = list(nfa[keys\_list[0]].keys())  
  
dfa[keys\_list[0]] = {}  
for y in range(t):  
 var = "".join(nfa[keys\_list[0]][path\_list[y]])  
 dfa[keys\_list[0]][path\_list[y]] = var  
 if var not in keys\_list:  
 new\_states\_list.append(var)  
 keys\_list.append(var)  
  
while len(new\_states\_list) != 0:  
 dfa[new\_states\_list[0]] = {}  
 for \_ in range(len(new\_states\_list[0])):  
 for i in range(len(path\_list)):  
 temp = []  
 for j in range(len(new\_states\_list[0])):  
 temp += nfa[new\_states\_list[0][j]][path\_list[i]]  
 s = ""  
 s = s.join(temp)  
 if s not in keys\_list:  
 new\_states\_list.append(s)  
 keys\_list.append(s)  
 dfa[new\_states\_list[0]][path\_list[i]] = s  
  
 new\_states\_list.remove(new\_states\_list[0])  
  
print("\nDFA :- \n")  
print(dfa)  
print("\nPrinting DFA table :- ")  
dfa\_table = pd.DataFrame(dfa)  
print(dfa\_table.transpose())  
  
dfa\_states\_list = list(dfa.keys())  
dfa\_final\_states = []  
for x in dfa\_states\_list:  
 for i in x:  
 if i in nfa\_final\_state:  
 dfa\_final\_states.append(x)  
 break

**OUTPUT:**

****

****

**RESULT:**

* The given NFA was converted to a DFA using Python successfully.