

# 18CSC304J/ Compiler Design

Submitted By:- ANANNYA P. NEOG (RA1911003010367)

## Exp-3: Conversion from NFA to DFA

**Aim:-** To convert a NFA to DFA

### **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

print("\nFinal states of the DFA are : ", dfa_final_states)

```

```

1 import pandas as pd
2
3 nfa = {}
4 n = int(input("No. of states : "))
5 t = int(input("No. of transitions : "))
6 for i in range(n):
7     state = input("state name : ")
8     nfa[state] = {}
9     for j in range(t):
10        path = input("path : ")
11        print("Enter end state from state {} travelling through path {} : ".format(state, path))
12        reaching_state = [x for x in input().split()]
13        nfa[state][path] = reaching_state
14
15 print("\nNFA :- \n")
16 print(nfa)
17 print("\nPrinting NFA table :- ")
18 nfa_table = pd.DataFrame(nfa)
19 print(nfa_table.transpose())
20
21 print("Enter final state of NFA : ")
22 nfa_final_state = [x for x in input().split()]
23
24 new_states_list = []
25
26 #-----
27
28 dfa = {}
29 keys_list = list(
30     list(nfa.keys())[0])
31 path_list = list(nfa[keys_list[0]].keys())
32
33 dfa[keys_list[0]] = {}
34 for y in range(t):
35     var = "".join(nfa[keys_list[0]][
36         path_list[y]])
37     dfa[keys_list[0]][path_list[y]] = var
38     if var not in keys_list:
39         new_states_list.append(var)
40         keys_list.append(var)
41

```

```

main.py
41
42 while len(new_states_list) != 0:
43     dfa[new_states_list[0]] = {}
44     for _ in range(len(new_states_list[0])):
45         for i in range(len(path_list)):
46             temp = []
47             for j in range(len(new_states_list[0])):
48                 temp += nfa[new_states_list[0][j]][path_list[i]]
49             s = ""
50             s = s.join(temp)
51             if s not in keys_list:
52                 new_states_list.append(s)
53                 keys_list.append(s)
54             dfa[new_states_list[0]][path_list[i]] = s
55
56     new_states_list.remove(new_states_list[0])
57
58 print("\nDFA :- \n")
59 print(dfa)
60 print("\nPrinting DFA table :- ")
61 dfa_table = pd.DataFrame(dfa)
62 print(dfa_table.transpose())
63
64 dfa_states_list = list(dfa.keys())
65 dfa_final_states = []
66 for x in dfa_states_list:
67     for i in x:
68         if i in nfa_final_state:
69             dfa_final_states.append(x)
70             break
71
72 print("\nFinal states of the DFA are : ", dfa_final_states)

```

## Output:-

```

Input
No. of states : 3
No. of transitions : 2
state name : A
path : 0
Enter end state from state A travelling through path 0 :
A
path : 1
Enter end state from state A travelling through path 1 :
AB
state name : B
path : 0
Enter end state from state B travelling through path 0 :
C
path : 1
Enter end state from state B travelling through path 1 :
C
state name : C
path : 0
Enter end state from state C travelling through path 0 :
C
path : 1
Enter end state from state C travelling through path 1 :
C

NFA :-
{'A': {'0': ['A'], '1': ['AB']}, 'B': {'0': ['C'], '1': ['C']}, 'C': {'0': [], '1': []}}

Printing NFA table :-
    0  1
A  [A] [AB]
B  [C]  [C]
C  []  []

Enter final state of NFA :
C

DFA :-
{'A': {'0': 'A', '1': 'AB'}, 'AB': {'0': 'AC', '1': 'ABC'}, 'AC': {'0': 'A', '1': 'AB'}, 'ABC': {'0': 'AC', '1': 'ABC'}}

Printing DFA table :-
    0  1
A    A  AB
AB  AC  ABC
AC    A  AB
ABC  AC  ABC

Final states of the DFA are :  ['AC', 'ABC']

...Program finished with exit code 0
Press ENTER to exit console.

```