

18CSC304J Compiler Design Lab

Exercise 3: Conversion from NFA to DFA

**Submitted To:-
Dr.M.Kanchana**

**Submitted By:-
Name:- Puneet Sharma
Reg.No. :- RA1911003010331**

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)

```

OUTPUT:-

```

No. of states : 3
No. of transitions : 2
state name : 2
path : 0
Enter end state from state 2 travelling through path 0 :
2
path : 1
Enter end state from state 2 travelling through path 1 :
2
state name : 1
path : 0
Enter end state from state 1 travelling through path 0 :
1
path : 1
Enter end state from state 1 travelling through path 1 :
2
state name : 0
path : 0
Enter end state from state 0 travelling through path 0 :
1
path : 1
Enter end state from state 0 travelling through path 1 :
1

NFA :-

{'2': {'0': ['2'], '1': ['2']}, '1': {'0': ['1'], '1': ['2']}, '0': {'0': ['1'], '1': ['1']}}

Printing NFA table :-
    0    1
2  [2]  [2]
1  [1]  [2]
0  [1]  [1]
Enter final state of NFA :
2

DFA :-

{'2': {'0': '2', '1': '2'}}

Printing DFA table :-
    0    1
2  2    2

Final states of the DFA are :  ['2']

```