

Exp 4 - NFA to DFA

Aim: To convert a given NFA into DFA

Program:

```
import pandas as pd
# Taking NFA input from User
nfa = {}
n = int(input("No. of states : ")) #Enter total no. of states
t = int(input("No. of transitions : ")) #Enter total no. of
transitions/paths eg: a,b so input 2 for a,b,c input 3
for i in range(n):
    state = input("state name : ") #Enter state name eg: A, B, C, q1,
q2 ..etc
    nfa[state] = {} #Creating a nested dictionary
    for j in range(t):
        path = input("path : ") #Enter path eg : a or b in {a,b} 0 or 1 in
{0,1}
        print("Enter end state from state {} travelling through path {} : ".format(state,path))
        reaching_state = [x for x in input().split()] #Enter all the end
states that
        nfa[state][path] = reaching_state #Assigning the end states to the
paths in dictionary

print("\nNFA :- \n")
print(nfa) #Printing 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()] # Enter final state/states
of NFA
#####
new_states_list = [] #holds all the new states created in dfa
dfa = {} #dfadictionary/table or the output structure we needed
keys_list = list(list(nfa.keys())[0]) #contains all the states in nfa plus
the states created in dfa are also appended further
path_list = list(nfa[keys_list[0]].keys()) #list of all the paths eg: [a,b]
or [0,1]
#####
# Computing first row of DFA transition table
dfa[keys_list[0]] = {} #creating a nested dictionary in dfa
for y in range(t):
    var = "".join(nfa[keys_list[0]][path_list[y]]) #creating a single
string from all the elements of the list which is a new state
    dfa[keys_list[0]][path_list[y]] = var #assigning the state in DFA table
    if var not in keys_list: #if the state is newly created
```

Exp 4 - NFA to DFA

RA1911003010143

Abhishek Kumar

```
        new_states_list.append(var) #then append it to the new_states_list
        keys_list.append(var) #as well as to the keys_list which contains
all the states
#####
# Computing the other rows of DFA transition table
while len(new_states_list) != 0: #condition is true only if the
new_states_list is not empty
    dfa[new_states_list[0]] = {} #taking the first element of the
new_states_list and examining it
    for _ in range(len(new_states_list[0])):
        for i in range(len(path_list)):
            temp = [] #creating a temporary list
            for j in range(len(new_states_list[0])):
                temp += nfa[new_states_list[0][j]][path_list[i]] #taking
the union of the states
            s = ""
            s = s.join(temp) #creating a single string(new state) from all
the elements of the
            if s not in keys_list: #if the state is newly created
                new_states_list.append(s) #then append it to the
new_states_list
                keys_list.append(s) #as well as to the keys_list which
contains all the states
            dfa[new_states_list[0]][path_list[i]] = s #assigning the new
state in the DFA table
            new_states_list.remove(new_states_list[0]) #Removing the first element
in the new_states_list
print("\nDFA :- \n")
print(dfa) #Printing the DFA created
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) #Printing Final
states of DFA
```

Exp 4 - NFA to DFA
RA1911003010143
Abhishek Kumar

Output:

```
exp4.py - Untitled (Workspace) - Visual Studio Code

PROBLEMS OUTPUT DEBUG CONSOLE JUPYTER TERMINAL

PS C:\Users\abhis\Desktop\Web Development\Back-end\Node-udemy-course\web-server> python -u "c:\Users\abhis\Desktop\Study\6th Semester\Compiler Design\Lab\Exp4\exp4.py"
No. of states : 4
No. of transitions : 2
state name : a
path : 0
Enter end state from state a travelling through path 0 :
a b
path : 1
Enter end state from state a travelling through path 1 :
a c
state name : b
path : 0
Enter end state from state b travelling through path 0 :
d
path : 1
Enter end state from state b travelling through path 1 :

state name : c
path : 0
Enter end state from state c travelling through path 0 :

path : 1
Enter end state from state c travelling through path 1 :
d
state name : d
path : 0
Enter end state from state d travelling through path 0 :

path : 1
Enter end state from state d travelling through path 1 :

NFA :-

{'a': {'0': ['a', 'b'], '1': ['a', 'c']}, 'b': {'0': ['d'], '1': []}, 'c': {'0': [], '1': ['d']}, 'd': {'0': [], '1': []}}

Printing NFA table :-
      0      1
a [a, b] [a, c]
b [d]    []
c []     [d]
d []     []
Enter final state of NFA :
d

DFA :-

{'a': {'0': 'ab', '1': 'ac'}, 'ab': {'0': 'abd', '1': 'ac'}, 'ac': {'0': 'ab', '1': 'acd'}, 'abd': {'0': 'abd', '1': 'ac'}, 'acd': {'0': 'ab', '1': 'acd'}}

Printing DFA table :-
      0      1
a   ab   ac
ab  abd  ac
ac   ab  acd
abd  abd  ac
acd  ab  acd

Final states of the DFA are : ['abd', 'acd']
PS C:\Users\abhis\Desktop\Web Development\Back-end\Node-udemy-course\web-server>
```

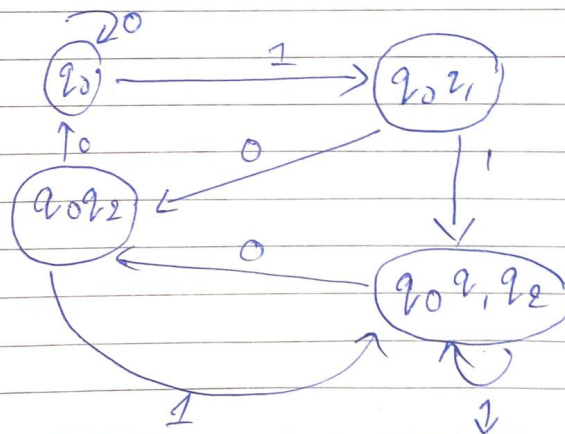
Manual Calculation:

Compiler design Exp-4

PAGE NO.:
DATE: / /

States	0	1
q_0	q_0	q_0, q_1
q_1	q_2	q_2
q_2	—	q_2

States	0	1
q_0	q_0	q_0, q_1
$q_0 q_1$	$q_0 q_2$	$q_0 q_1, q_2$
$q_0 q_2$	q_0	$q_0 q_1, q_2$
$q_0 q_1 q_2$	$q_0 q_2$	$q_0 q_1 q_2$



Exp 4 - NFA to DFA

RA1911003010143

Abhishek Kumar

Result: Program to convert NFA to DFA was written and executed successfully.