10-Feb-22

Experiment 3 - NFA to DFA conversion

Dhawal Patil
RA1911003010575
CSE A2

Aim:

Write A Program to convert given NFA to DFA.

Algorithm:

Step 1) Get the input from the user
Step 2) Set the only state in SDFA to “unmarked”.
Step 3) While SDFA contains an unmarked state do -
  a) Assume T is that unmarked state
  b) For each a in % do S = e-closure(MoveNFA(T,a))
  c) If S is not SDFA already then, add S to SDFA (as unmarked state)
  d) Set MoveDFA(T,a) to S
Step 4) For each S in SDFA if any s & S is a final state in the NFA then, mark S as final state in the DFA
Step 5) Print the result

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:

```
NFA :-

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

Printing NFA table :-
     0       1
A  [A]   [A, B]
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.
```

Result:

Hence conversion of NFA to DFA was successfully completed and the desired result was obtained.