Assignment-1

AUTOMATA - THEORY

Aim :- Write a python script to convert a NFA to DFA.

Functions used in the script :

-> ***def final_accept_states(dfa_final,nfa_final)*** *:*

This function is used to generate the final accept states of the resulting DFA.

It takes two inputs " dfa_final " and " nfa_final " and the input from " nfa_final " is considered and checked in all the states of " dfa_final " and we get our final accepting states of new resulting DFA.

Formula : $FD = \{q \epsilon QD : FN \cap q \neq \phi\}$.

-> ***def generate_t_function(t_func_nfa,input_alphabet,PowerSet_dfa)*** *:*

This function is used to generate the " t_func " of DFA and returns " list ".

It takes three inputs " t_func_nfa ", " input_alphabet ", " PowerSet_dfa ". This function finds the possible transition states of DFA from the t_func of NFA.

-> ***def make_PowerSet(set,set_size)*** :

This function is used to generate the " PowerSet "(i.e all possible states of DFA) of and the " States " for DFA and results in a list.

It takes two input " set "(i.e range of set) and " set_size"(i.e no. of states in nfa).

-> **In MAIN FUNCTION** :

Reads the input from the input.json (i.e : **with open('input.json', 'r') as f:**
nfa = json.load(f)).

And the called the above functions to get the " NEW states, letters, t_function, start and final " of DFA.

All the NEW states, letters, t_function, start and final is DUMPED in the Output.json file.
(i.e : **with open('output.json', 'w') as outfile:**
json.dump(dfa, outfile, indent=3)).

SAMPLE INPUT :[input.json]

```
 {
"states" : 2,
"letters" : ["a","b"],
"t_func" : [[1,"a",[0,1]]],
"start" : 0,
"final" : [1]
}
```

SAMPLE OUTPUT: [output.json]

```
{
"states": 4,
"letters": [
   "a",
   "b"
],
"t_func": [
   [
      [],
      "a",
  []
],
[
  [],
  "b",
  []
],
[
  [
    0
  ],
  "a",
  []
],
[
  [
    0
  ],
  "b",
  []
],
[
  [
    1
  ],
  "a",
  [
    0,
    1
  ]
],
[
  [
    1
  ],
  "b",
  []
],
[
  [
    0,
    1
  ],
  "a",
  [
    0,
    1
  ]
],
[
  [
    0,
        1
      ],
      "b",
      []
    ]
  ],
  "start": 0,
  "final": [
    [
      1
    ],
    [
      0,
      1
    ]
  ]
```

ASSUMPTIONS :
1. There are no NULL transitions
2. Input.json and Output.json is already present in folder.
3. All the transitions, including those that have no next state, is included in the t_func of the DFA