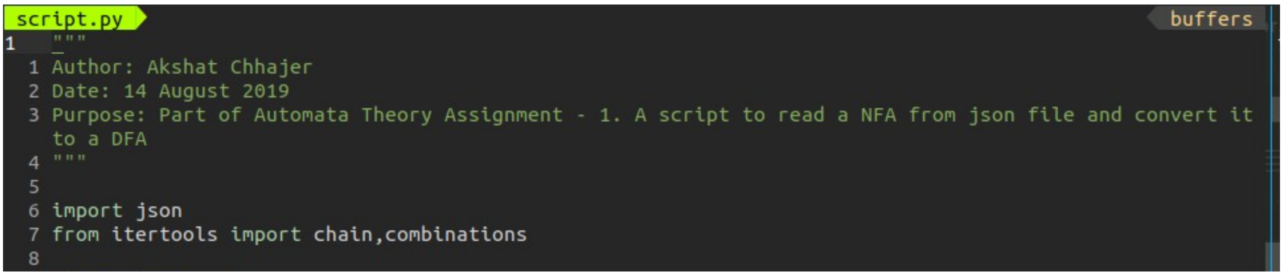
Akshat Chhajer
2018114008

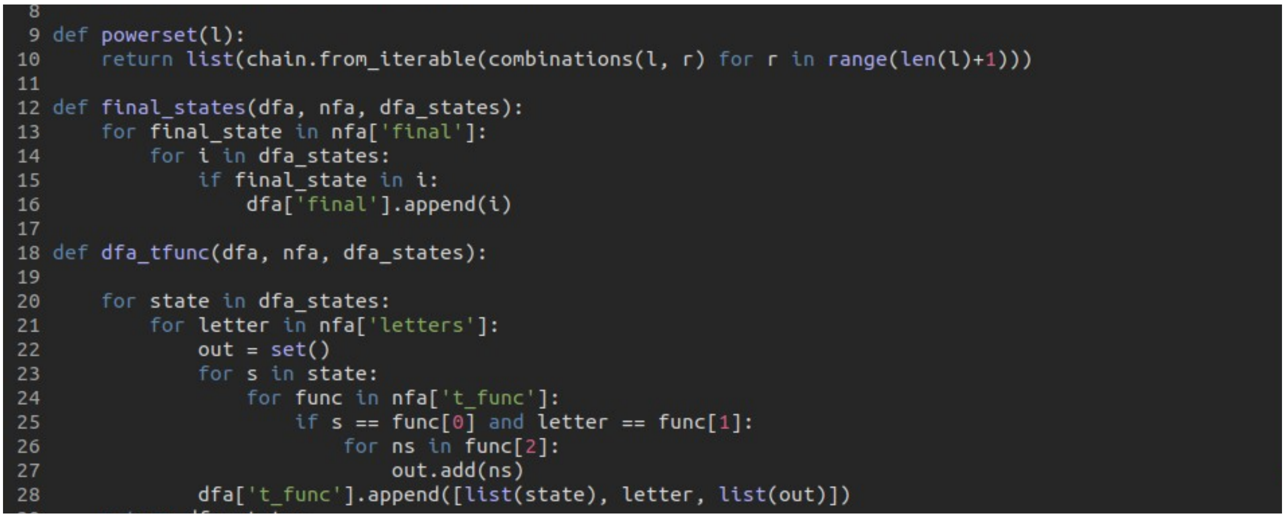
Automata Theory

Assignment 1 – Code Report



```
"""
Author: Akshat Chhajer
Date: 14 August 2019
Purpose: Part of Automata Theory Assignment - 1. A script to read a NFA from json file and convert it to a DFA
"""

import json
from itertools import chain,combinations

```

Basic information about the code and the imports



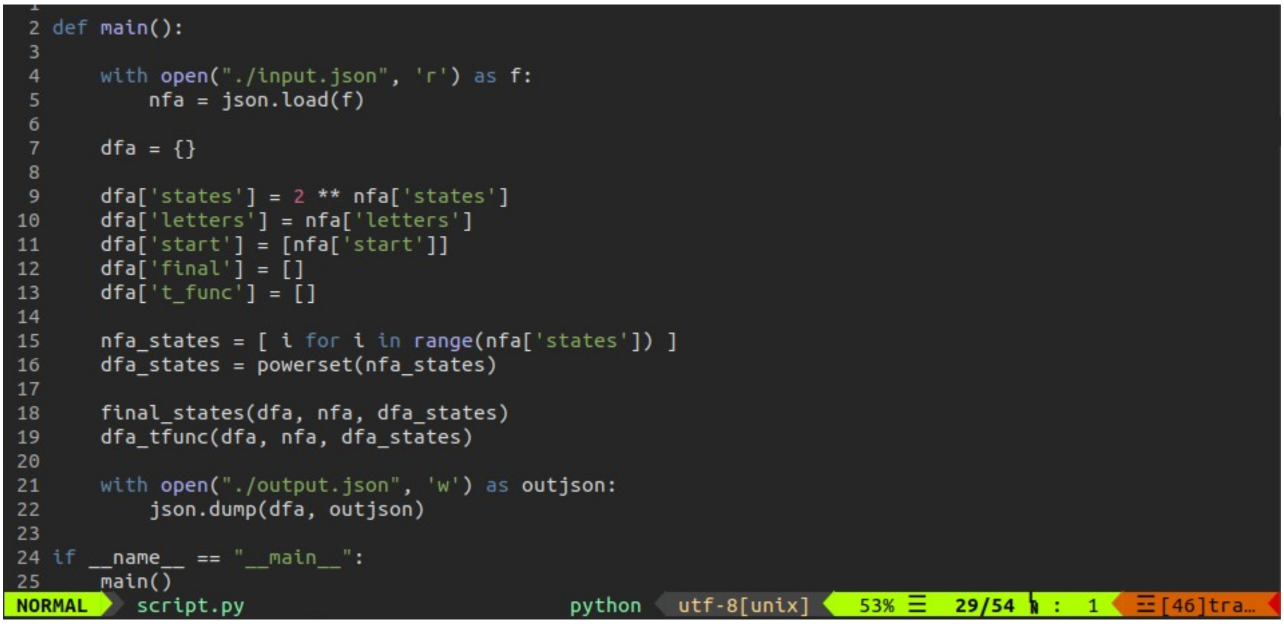
```

def powerset(l):
    return list(chain.from_iterable(combinations(l, r) for r in range(len(l)+1)))

def final_states(dfa, nfa, dfa_states):
    for final_state in nfa['final']:
        for i in dfa_states:
            if final_state in i:
                dfa['final'].append(i)

def dfa_tfunc(dfa, nfa, dfa_states):

    for state in dfa_states:
        for letter in nfa['letters']:
            out = set()
            for s in state:
                for func in nfa['t_func']:
                    if s == func[0] and letter == func[1]:
                        for ns in func[2]:
                            out.add(ns)
            dfa['t_func'].append([list(state), letter, list(out)])
```

Utility Functions:

1. **powerset(l)** – Function takes in a list and returns the powerset of the elements of that list

2. **final_states(dfa, nfa, dfa_states)**
Function takes in the current dfa, nfa, dfa_states and adds the expected final states to the dfa. This is done by iterating over the dfa_states and selecting the ones which include any of the nfa final states.

3. dfa_tfunc(dfa, nfa, dfa_states)
Function takes in the the current dfa, nfa, and dfa_states and adds the corresponding transition functions to the dfa. This is done by maintaining a set of all the output states for each combination of states and input letters and adding it to ‘t_func’



```
 2 def main():
 3 
 4     with open("./input.json", 'r') as f:
 5         nfa = json.load(f)
 6 
 7     dfa = {}
 8 
 9     dfa['states'] = 2 ** nfa['states']
10     dfa['letters'] = nfa['letters']
11     dfa['start'] = [nfa['start']]
12     dfa['final'] = []
13     dfa['t_func'] = []
14 
15     nfa_states = [ i for i in range(nfa['states']) ]
16     dfa_states = powerset(nfa_states)
17 
18     final_states(dfa, nfa, dfa_states)
19     dfa_tfunc(dfa, nfa, dfa_states)
20 
21     with open("./output.json", 'w') as outjson:
22         json.dump(dfa, outjson)
23 
24 if __name__ == "__main__":
25     main()
```

Main Function

- First, the given input.json is read from the same folder as a dictionary named nfa (4-5)
- nfa dictionary is initialized with the default values using the nfa (7-13)
- Next, the list of nfa_states and dfa_states is calculated using the powerset function (15-16)
- Then, final states and t_func of dfa are updated using the utility functions (18-19)
- Finally the output.json is written consisting of the final dfa. (21-22)