Roll No : 2018113003

# Report

**Convert a non-deterministic finite automata to a deterministic finite automata**

## Solution

### States

- The total states are Pow(Q) and thus the number of states in the DFA will be $2^Q$.

### Letters

- The letters do not change from an NFA to a DFA and hence no change

### Delta function

- The delta function is as follows:

```
## The Delta Function
t = []
for i in pq:
    for j in data['letters']:
        nex_st = set()
        for k in data['t_func']:
            if k[0] in i and j == k[1]:
                nex_st.update(k[2])
        t.append([list(i),j,list(nex_st)])
```

    - Where :
        - t → Final Delta function
        - pq → The powerset of Q
        - Nex_st → The next state going from input on an alphabet
- The delta function needs to find all the transitions from the input states to output states
- Hence, iterating over all the input states. Also the mapping via all the letters need to be known, hence the iteration over all the letters, and finally to check whether the transition state occurs or not, hence

iterating over all the transitions in the NFA, thus giving the required DFA

### Initial States

- Looping over all the states, find the states such that $i(NFA) \in$ Current State

### Final States

- Looping over all the states, find the states such that for all elements in $F(NFA) \in$ Current State