AUTOMATA THEORY

Converting NFA to DFA

This script converts Nondeterministic Finite Automaton to a Deterministic Finite Automaton.This script reads input.json file containing an object which represents a NFA. This json object will have 5 key-value pairs corresponding to the 5-tuple used to represent the NFA. A sample is shown below for input.json:

```
{
    "states": 8,
    "letters": ["a", "b", "c"],
    "t_func" : [1, 'a', [1,3,0]],
    "start" : 0,
    "final" : [4]
}
```

- states : Number of states. Assume the states are numbered 0,1,2…n-1 for n states.
- letters: Alphabet used by the NFA
- t func: The transition function for the NFA. Each transition is an array of 3 elements: original state, input and the new state.
- start: The index of the starting state
- final: List of accepted states

Running the script

First install python3 and then install jsbeautifer

- $ pip3 install jsbeautifier
- $ python3 script.py

Explaining the code

open()

- ‘open’ function opens the input.json file with read only mode ‘r’ and returns file object.

json.loads()

- ‘json.load’ takes file object, and optional argument object_pairs_hook.
- The return value of object_pairs_hook will be used instead of the dict.

OrderedDict()

- OrderedDict preserves the order in which the keys are inserted

Indexing of States of DFA

- States of DFA are actually the sets of states of NFA.
- Total states of NFA is n from 0 to n-1. So total states of DFA is 2^n from 0 to 2^n-1.
- So if one state of DFA is [1,0,2] then it will be indexed as 2^1 + 2^0 + 2^2 = 7. So 7 represents the set [1,0,2]

finalState(l ,r, x, y)

- This function calculates all the set of states which contain atleast one final state of NFA to get the final states of DFA.
- Initially l=0 and r=NFA[“states”]-1, x = 0 and y = 0
- y = 1 means that atleast one final state is included in the final state of DFA.
- if y == 1 then x is the final state of DFA and is appended to DFA[“final”] list.
- First time, we call finalState(l+1, r, x, y) without adding lth state in the final state of DFA
- Then we call finalState(l+1, , x + 2^l, y) including l in the final state of DFA
- Y is the variable which contains all the final state of NFA i.e. Y’s jth bit is 1 if j is the final state of NFA.
- Then if (2^l & Y) != 0 which means that l is the final state of NFA then y is set to 1 otherwise y remains what it is.

t_func(DFA, LFA, L)

- This function returns DFA[“t_func”]
- L = no of alphabets in DFA[“letters”]
- L*i+j th member of DFA[“t_func”] list represents that ‘ith’ state is given ‘alphabet DFA[“letters”][j]’ and returns ‘jth’ state
- Firstly, DFA[“t_func”] is initialized with 0
- Then all the mappings in NFA[“t_func”] is added to DFA[“t_func”] after indexing them.
- Then for each i from 0 to 2^NFA[“states”]-1, we know that i represents a set, so union of outputs of all the elements in a set is the output of i when DFA[“letter”][j] is given.

OUTPUT

- This script returns DFA equivalent of NFA as json object in output.json file.