**HW 4, CS 603**

**Xiaolong Sun**

**Design and Implementation**

In the documentation, I will first go through some key data structure for NFA and DFA. Then, I will discuss two functions that play important roles in NFA to DFA conversion. Lastly, I will put everything together and present the main function.

The actual implementation is done by Python.

1. **Data Structure for NFA**

We use dictionary to store an NFA:

Nfa = {current\_state:[[output\_state1, output\_state2,… ], # ouput state for input 0

[output\_state\_a, output\_state\_b,…] # ouput state for input 1

]}

As the pseudo code above shows, the key of the dictionary represents the input state. The value is a list of two elements. The first element is a list of output states when the input is 0. The second element is a list of output states when the input is 1.

Using this definition, the NFA from HW4 on Canvas will be:

nfa1 = {'0':[['0','1'], ['3']],\

'1':[['0'], ['1','3']],\

'2':[[''],['0','2']],\

'3':[['0','1','2'],['1']]

}

1. **Data Structure for DFA**

The way to define DFA is similar to that of NFA. For example, the corresponding DFA for nfa1 can be defined as:

Dfa = {'0': [['01'], ['3']],

'01': [['01'], ['13']],

'3': [['012'], ['1']],

'13': [['012'], ['13']],

'012': [['01'], ['0123']],

'1': [['0'], ['13']],

'0123': [['012'], ['0123']]}

1. **Core Functions**

There are two functions needs for implementing the conversion.

The first function is called *merge\_state(list\_states:list)*. For given state and input, if there are multiple output states. For example, ['0','1'] --> ['01']. The concatenated states will be returned.

The second function is called *get\_dfa\_output\_state(current\_state, input\_string ,nfa)*. Given the current state and input, find all output\_state for dfa from nfa.

1. **Driver Script and Conversion Algorithm**

The driver script starts with the initial state. Then it uses a queue to store states that have not been visited yet.

While the queue is not empty, the driver script will visit each state in the queue and uses *get\_dfa\_output\_state()* to get corresponding output for input 0 and for input 1 on that state.

Finally, the dfa will be printed.