Assignment #3A and # 3B
Submission Deadline: March 5, 2018
Maximum Marks: 40

# **3A** In this programming assignment you will implement ϵ-NFA to DFA conversion mechanishm. The specific input-output requirements to your program are given below:

Input: An ϵ-NFA N_ϵ with alphabet $\Sigma = \{0, 1\}$. The description of the NFA must be provided to the program through a file. An example format for a specific ϵ-NFA is given below. The first row includes all the states; the second row include the starting state; the third row includes all the final states; fourth row describes three transitions from q_0 on input symbols $0, 1$ and ϵ respectively; fifth row does the same with respect to q_1; and so on.

q_0	q_1	q_2	q_3
q_0			
q_1			
$\{q_1\}$	ϕ	$\{q_1, q_2\}$	
$\{q_1, q_2\}$	$\{q_1\}$	$\{q_3\}$	
$\{q_1\}$	ϕ	ϕ	
$\{q_2\}$	ϕ	$\{q_2\}$	

Output: An equivalent DFA D, i.e. $\mathcal{L}(D) = \mathcal{L}(N_\epsilon)$. The output, i.e., the description (similar to input NFA description) of equivalent DFA D must be copied into a file.

Your program must also provide a graphical drawing of the equivalent DFA (you may use standard libraries for this task).

# **3B** In this programming assignment you will implement membership checking of an arbitrary binary string with respect to the language of a given ϵ-NFA. The specific input-output requirements to your program are given below:

Input: An ϵ-NFA N_ϵ with alphabet $\Sigma = \{0, 1\}$ and a string $w \in \{0, 1\}^*$. The input N_ϵ will be provided in the format given in Assignment # 3A.

Output: Yes, if $w \in \mathcal{L}(N_\epsilon)$; No - otherwise.

Important Instructions!!

- Combine both 3A and 3B in one program.
- 20 marks for correctness of # 3A.
- 10 marks for correctness of # 3B.
- The rest 10 will account for programming efficiency and proficiency.

Assignment #3C
Submission Deadline: March 12, 2018
Maximum Marks: 30

In this programming assignment you will implement membership checking of an arbitrary string with respect to the language of a given regular expression. The specific input-output requirements to your program are given below:

Input: A Regular expression R over a alphabet Σ, and a string $w \in \Sigma^*$. The Σ need not be $\{0, 1\}$. Thus, as input, you must also provide the alphabet Σ.

Output: Yes, if $w \in \mathcal{L}(R)$; No - otherwise.

Your program should ideally be doing the following:

- Function making sense of regular expressions.
- Function that converts regular expressions to its equivalent ϵ-NFA N_ϵ.
- Using your # 3B program (modify it to support any alphabet Σ) to check if $w \in \mathcal{L}(N_\epsilon)$.

Note: 25 marks for correctness of # 3C. The rest 5 will account for programming efficiency and proficiency.