Homework 3

CSE 105, Fall 2025
Due: Monday, October 20, 11:59 pm

Your assignments in this class will be evaluated not only on the correctness of your answers, but on your ability to present your ideas clearly and logically. You should always explain how you arrived at your conclusions, using mathematically sound reasoning, where applicable. Whether you use formal proof techniques or write a more informal argument for why something is true, your answers should always be well-supported. Your goal should be to convince the reader that your results and methods are sound.

Instructions:

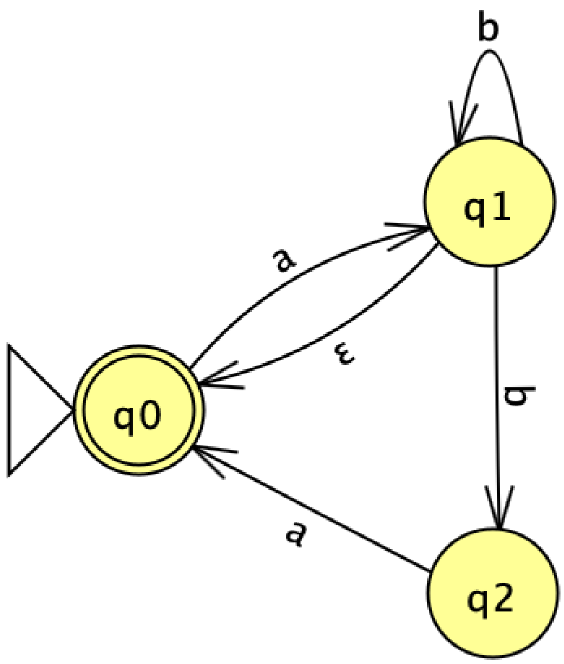
- Read each question carefully.
- Selection of questions
  - You are required to answer <u>one question from each topic</u> in this homework.
  - You may choose any one question from the set of questions within each topic.
- Each question is worth 5 points (25 points total).
- Submission guidelines:
  - Upload a single file to Gradescope for each group.
  - All group members' names and PIDs should be on each page of the submission.
  - Include the text of each question you are answering, followed by your solution.

Topics:

1. Converting an NFA to a DFA
2. Languages Described by Regular Expressions
3. Converting from Regular Expressions to Automata
4. Converting from Automata to Regular Expressions
5. Proving Closure of Regular Operations Using Regular Expressions

Problem 1:

Given the following NFA, create a DFA that recognizes the same language using subset construction by giving the final state diagram of the DFA.



Problem 2:

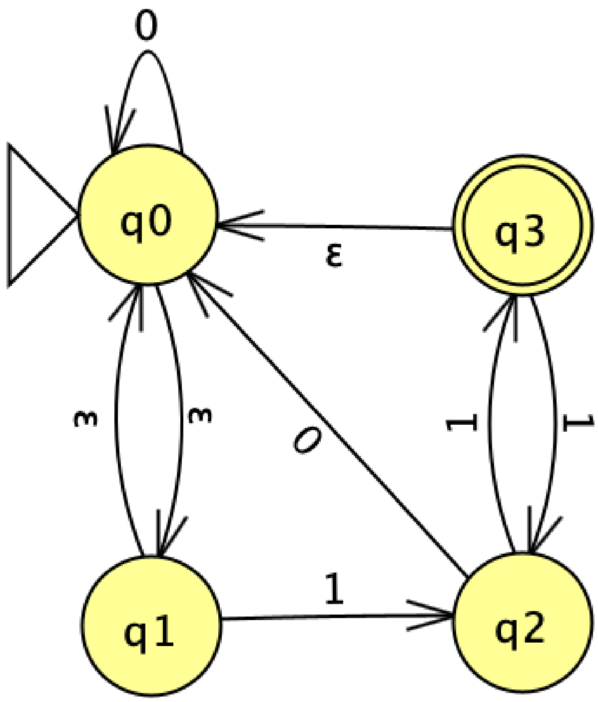
Suppose we have the NFA N defined as follows: $N = (\{NEW, E, O, S, F\}, \{0, 1\}, \delta, NEW, \{E, F\})$ where δ is given as:

	0	1	ε
NEW	$\varnothing$	$\varnothing$	$\{E, S\}$
E	$\{O\}$	$\{E\}$	$\varnothing$
O	$\{E\}$	$\{O\}$	$\varnothing$
S	$\{S\}$	$\{F\}$	$\varnothing$
F	$\{S\}$	$\{F\}$	$\varnothing$

Using subset construction, give the 5-tuple of a DFA D that recognizes the same language.

Problem 3:

Given the following NFA, create a DFA that recognizes the same language using subset construction by giving the final state diagram of the DFA.



Problem 4:

Suppose we have the NFA N defined as follows: $N = (\{q_0, q_1, q_2, q_3, q_4\}, \{0, 1\}, \delta, q_0, \{q_4\})$ where δ is given as:

	0	1	ε
q_0	$\{q_0\}$	$\{q_0, q_1\}$	$\varnothing$
q_1	$\{q_2\}$	$\{q_2\}$	$\varnothing$
q_2	$\{q_3\}$	$\{q_3\}$	$\varnothing$
q_3	$\varnothing$	$\varnothing$	$\{q_4\}$
q_4	$\varnothing$	$\varnothing$	$\varnothing$

Using subset construction, give the 5-tuple of a DFA D that recognizes the same language.

Problem 5:

Describe in words the language given by the regular expression $R = 1^{*}010^{*}$ over $\Sigma = \{0, 1\}$ and give 5 strings in the language and 5 strings not in the language.

Problem 6:

Given the language $L = \{w \mid w \text{ does not contain the substring } try\}$ over $\Sigma = \{q, w, e, r, t, y\}$, find a regular expression R that describes the same language.

Problem 7:

Give a regular expression R over $\Sigma = \{0, 1, 2\}$ that describes the language L over Σ containing only strings that start and end with the same symbol (empty string included).

Problem 8:

Describe in words the language given by the regular expression $R = (1^{*}0^{*})^{*}101(0^{*}1^{*})^{*}$ over $\Sigma = \{0, 1\}$ and give 5 strings in the language and 5 strings not in the language.

Problem 9:

Create an NFA that recognizes the same language as $L(R)$ where we have the regular expression
$R = (\varepsilon \cup 01 \cup 1)^* \cup 00(0^*1^*0^*)^*$

Problem 10:

Create an NFA that recognizes the same language as $L(((11)^*0^*1) \cup ((00)^*1^*0)^*)$ over $\Sigma = \{0, 1\}$.

Problem 11:

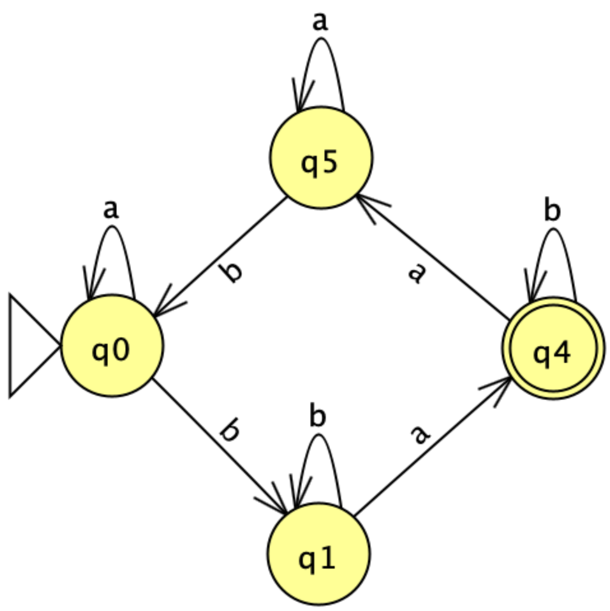
Given the language $L((a \cup c \cup d \cup b(a \cup b \cup d \cup c(b \cup c \cup d)))^*)$ over
$\Sigma = \{a, b, c, d\}$, create a **3-state NFA** that recognizes the same language.

Problem 12:

Create an NFA that recognizes the same language as $L((ad \cup bc)\, a^*(bb \cup dd)c^*)$ over
$\Sigma = \{a, b, c, d\}$.

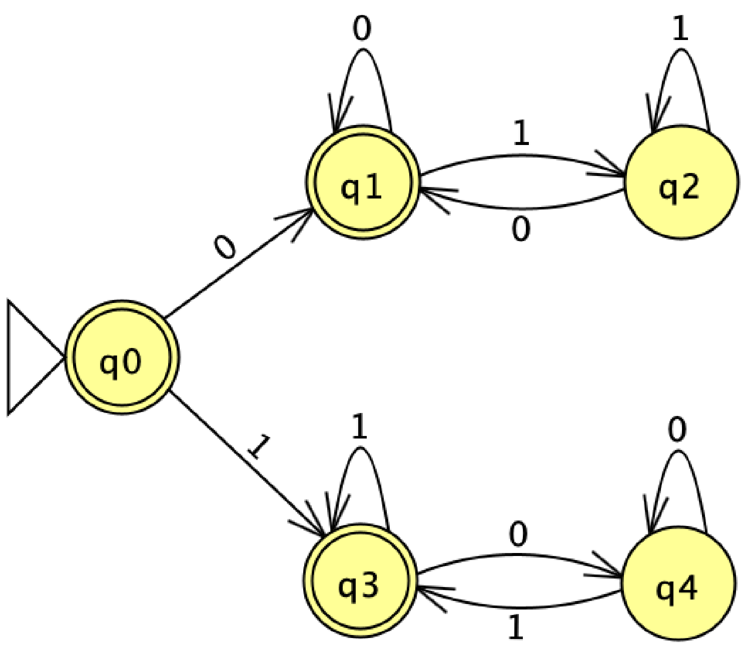
Problem 13:

Given the DFA D and D's state diagram given below, find the regular expression R such that $L(R) = L(D)$ by constructing a GNFA. Give all of the steps of your GNFA construction alongside R.



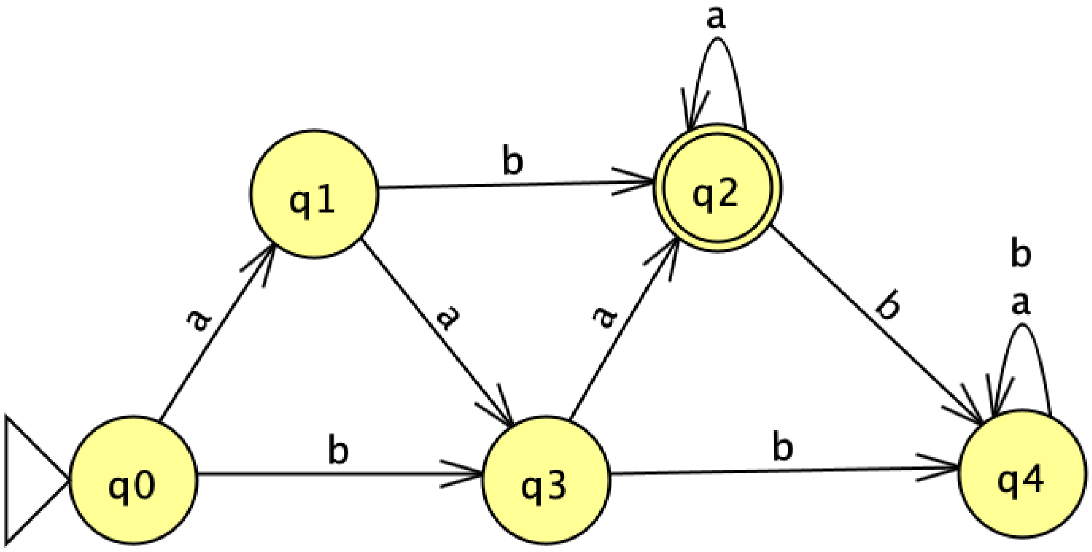
Problem 14:

Given the DFA D and D's state diagram given below, find the regular expression R such that $L(R) = L(D)$ by constructing a GNFA. Give all of the steps of your GNFA construction alongside R.



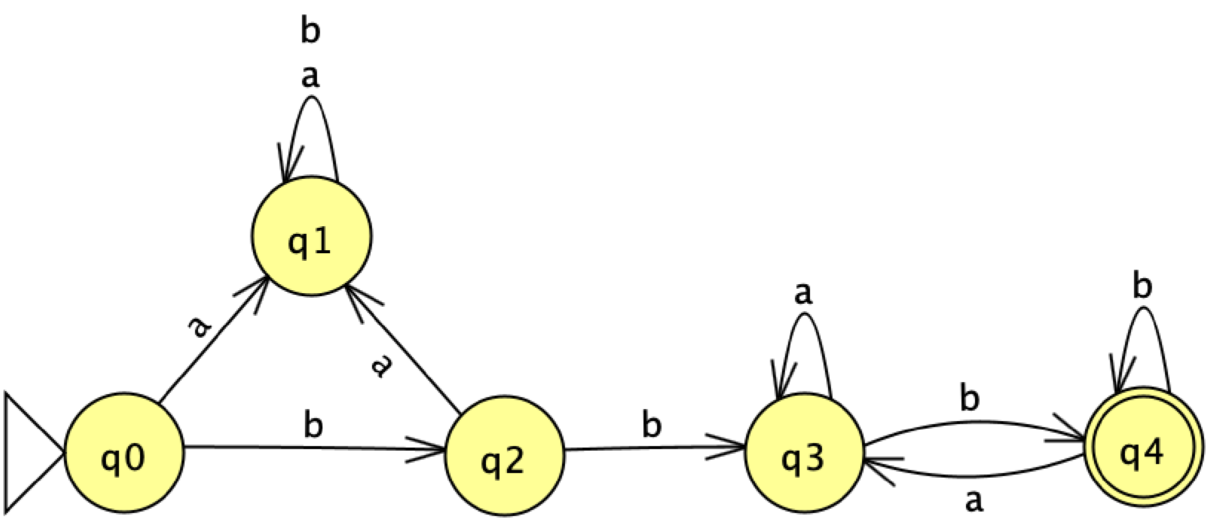
Problem 15:

Given the DFA D and D's state diagram given below, find the regular expression R such that $L(R) = L(D)$ by constructing a GNFA. Give all of the steps of your GNFA construction alongside R.



Problem 16:

Given the DFA D and D's state diagram given below, find the regular expression R such that $L(R) = L(D)$ by constructing a GNFA. Give all of the steps of your GNFA construction alongside R.



Problem 17:

Given a regular language L, recall that the reversal of the language is denoted as L^R , and it is the language of all the strings in L , but they are reversed. Given any regular expression R, prove that $L(R)^R$ is regular by constructing a regular expression R^R such that $L(R^R) = L(R)^R$.

Problem 18:

Given a regular expression R, we will define the even-length subset of $L(R)$ as $L_{even}(R) = \{w \in L(R) \mid |w| \textit{ is even}\}$. Show that $L_{even}(R)$ is regular by constructing a regular expression R_{even} such that $L(R_{even}) = L_{even}(R)$ for any given R.

Hint: Think about the complement regular expression, R_{odd} and how you can use it to help define R_{even}.

Problem 19:

Recall that p is a prefix of the string s if a string x exists such that $px = s$. Given any regular expression R, show that the set of prefixes of $L(R)$, denoted as $prefix(L(R))$, is regular. To do so, construct a regular expression R_{prefix} such that $L(R_{prefix}) = prefix(L(R))$ for any R.

Problem 20:

Recall that f is a suffix of the string s if a string x exists such that $xf = s$. Given any regular expression R, show that the set of suffixes of $L(R)$, denoted as $suffix(L(R))$, is regular. To do so, construct a regular expression R_{suffix} such that $L(R_{suffix}) = suffix(L(R))$ for any R.