CS 3823 - Theory of Computation: Homework Assignment 2

FALL 2025 **Due:** Friday, October 3, 2025

Related Reading. Chapter 1

Instructions. Near the top of the first page of your solutions please list clearly **all** the members of the group (please see the syllabus for the collaboration policy) who have created the solutions that you are submitting. Listing the names of the people in the group implies their full name and their 4x4 IDs. Alternatively, you can use the space below and provide the relevant information in case you submit the solutions using this document.

Student Information for the Solutions Submitted

	Lastname, Firstname	4x4 ID (e.g., dioc0000)
1	Khor, Arika	khor0006
2	Lawrence, Miles	lawr0039
3	Nguyen, Thuan	nguy0825
4	Williams, Cambren	will1394

Grade

Exercise	Pages	Your Score	Max
1	2		4
2	3		4
3	4-5		12
4	6-7		12
5	8		4
6	9		4
Total	2-9		40

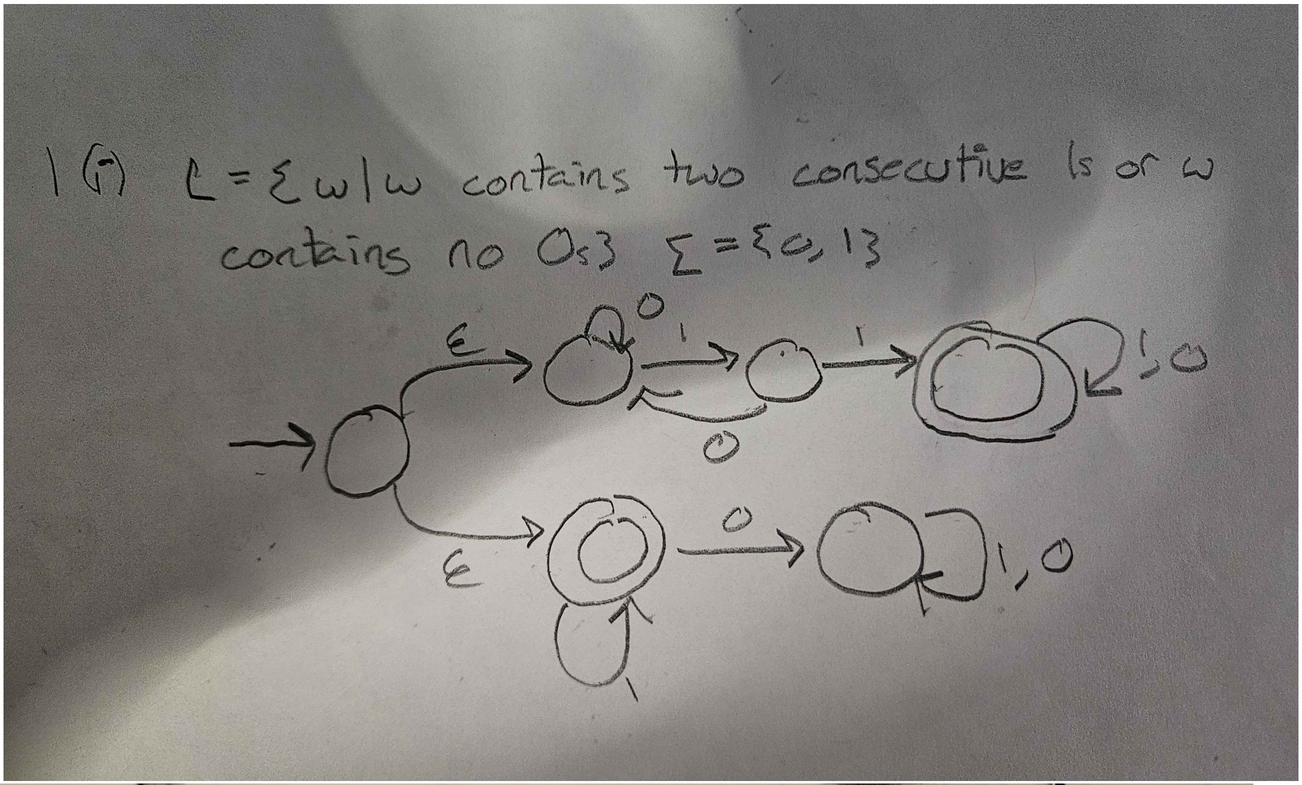
Additional Help and Resources. Did you use help and/or resources other than the textbook? Please indicate below.

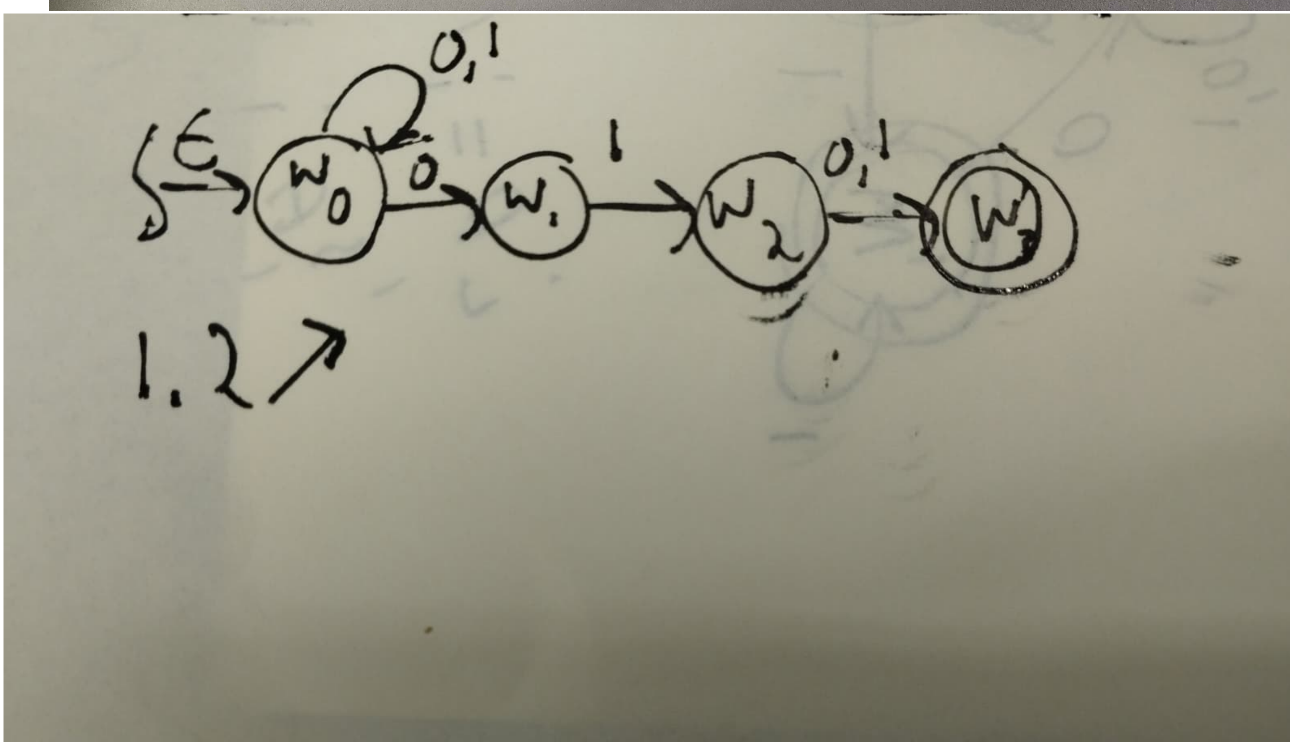
1 Design of NFAs [4 points; 2 points each]

Let $\Sigma = \{0, 1\}$. Give state diagrams for NFAs that recognize the following languages.

(i) $L_1 = \{w \mid w$ contains two consecutive 1s or w contains no 0s$\}$.

(ii) $L_2 = \{w \mid w = w_1 w_2 \ldots w_n$ such that $w_{n-3} = 0$ and $w_{n-2} = 1\}$.


1 (i) L = {w | w contains two consecutive 1s or w contains no 0s} Σ = {0, 1}
ε
0
1
1
1, 0
0
ε
0
1, 0
1


0, 1
ε
w0
0
w1
1
w2
0, 1
w3
1.2 ↗

2 NFAs and DFAs [8 points]

Consider the NFA $N = (\{q_1, q_2\}, \{0, 1\}, \delta, q_1, \{q_1\})$, with δ as defined below.

δ	0	1	ε
q_1	$\{q_1, q_2\}$	$\emptyset$	$\emptyset$
q_2	$\emptyset$	$\{q_1\}$	$\{q_1\}$

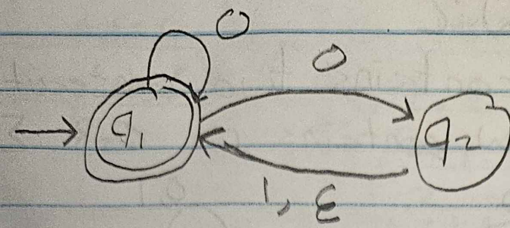
(i) **[2 points]** Draw the state diagram for N.

(ii) **[2 points]** What language does N recognize?

(iii) **[3 points]** Let M_1 be a DFA recognizing $L(N)$. Using the 'power set' construction that we saw in class for Theorem 2.2 of the book, draw the state diagram for M_1 with the corresponding members of $\mathcal{P}(\{q_1, q_2\})$.

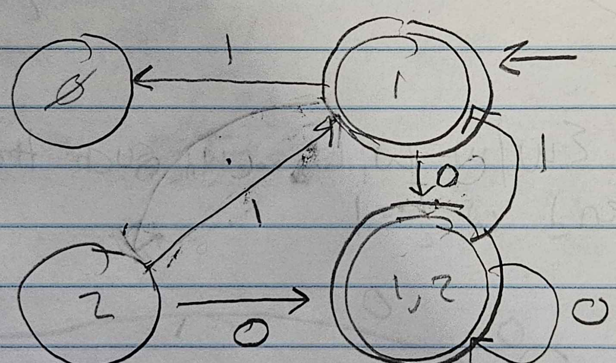
(iv) **[1 point]** Let M_2 be a DFA recognizing $L(M_1)$ but containing fewer states than M_1. Draw the state diagram of M_2.

2. (i)

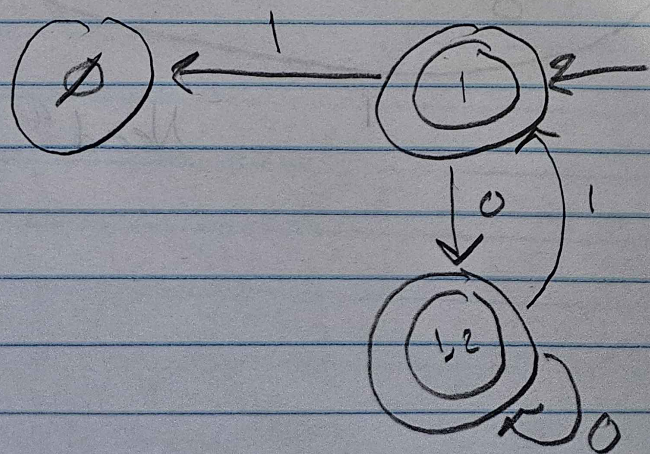


(ii) N is the langauge s.T. there is any combination 1s and 0s as long as there is never a substring 11 or begin with 1.

(iii)



(iv)



3 Representation [10 points]

(i) **[1 point]** Argue that if a language can be recognized by a DFA with k states, then it can be recognized by an NFA with k states.
A language can be recognized by a DFA with k states can also be recognized by an NFA with k states because a DFA is similar to an NFA.
To prove this, we can construct a NFA M = (Q1, Σ, δ_1, q_1, F_1) that recognizes a language by using a DFA N = (Q, Σ, δ, q_0, F) that also recognizes a language:

* M and N have the same set of states.
* Σ is the same as in N and M
* δ_1(r, a) = δ(r, a)
* $q_1 = q_0$
* F_1 = F

Since both DFA N and NFA M are similar and recognize the same language, then a DFA and a NFA with k states can recognize the same language.

Let $\Sigma = \{0, 1\}$. Let $\Sigma^n = \underbrace{\Sigma\Sigma\ldots\Sigma}_{n}$. Consider the regular language $L_k = \Sigma^*1\Sigma^{k-1}$ for some positive integer k.

(ii) **[2 points]** Show that L_k can be recognized by an NFA with $k+1$ states.
We can show that L_k can be recognized by an NFA with $k+1$ states by constructing a NFA N = (Q, Σ, δ, q_0, F) with k + 1 states that recognizes the language:

* Q = $\{q_0, ..., q_k\}$
* $\Sigma = \{0, 1\}$
* δ is as follows:

State	Input 0	Input 1
q_0	$\{q_0\}$	$\{q_0, q_1\}$
q_i $(1 \leqslant i \leqslant k-1)$	$\{q_{i+1}\}$	$\{q_{i+1}\}$
q_k	$\emptyset$	$\emptyset$

* $q_0 = \{q_0\}$
* F = $\{q_k\}$

(iii) **[4 points]** Prove that any DFA recognizing L_k must have at least 2^k states.
From the previous question, we showed that a NFA can recognize the language L_k with $k+1$ states. Therefore, we can construct a DFA out of the NFA, and with the powerset construction theorem, the total number of states that is in the DFA will be 2^{k+1} or $2^k * 2$, which proves that any DFA recognizing L_k has least 2^k states.

(iv) **[2 points]** What does part (iii) of this question tell you about the "powerset construction" theorem (Theorem 1.39 from Sipser's book)?
That any NFA with n states that recognizes a language can also be written as an equivalent DFA with at least 2^n states that recognizes the same language and anything less than 2^n states for a DFA would be impossible to construct out of a NFA.

(v) **[1 point]** What do parts (i), (ii), and (iii) of this question tell you about DFAs as compared to NFAs? That both DFAs and NFAs are very similar in how they recognize a language. However, NFAs are more easier to construct and are more powerful than DFAs.

4 Regular Expressions [4 points; 2 points each]

Let $\Sigma = \{0, 1\}$. Give regular expressions for the following languages.

(i) $L_1 = \{w \mid \text{every even position of } w = w_1 w_2 \ldots w_n \text{ has a 0}\}$.

$$\boxed{(\Sigma 0) * \Sigma}$$

Which simply translates to:

- LHS: (Right_input Left_input)
- Kliene star: * repeated n times (immediate left)
- RHS: Because LHS accepts 2 inputs, we need to deal with if $|L_1|$ is odd, so this accepts any string (that being Σ.

(ii) $L_2 = \{w \mid w \text{ interpreted as a binary number is divisible by 2 (or 10 in binary)}\}$.

Note that $\varepsilon \in L_1$, whereas $\varepsilon \notin L_2$.

$$\boxed{(\sum *0)}$$

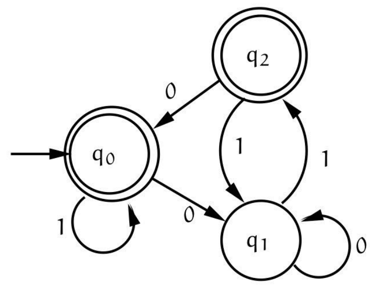
Meaning if it ends with a 0, its divisible by 2, given 2^n for n being even must be true

5 DFAs to Regular Expressions [6 points]

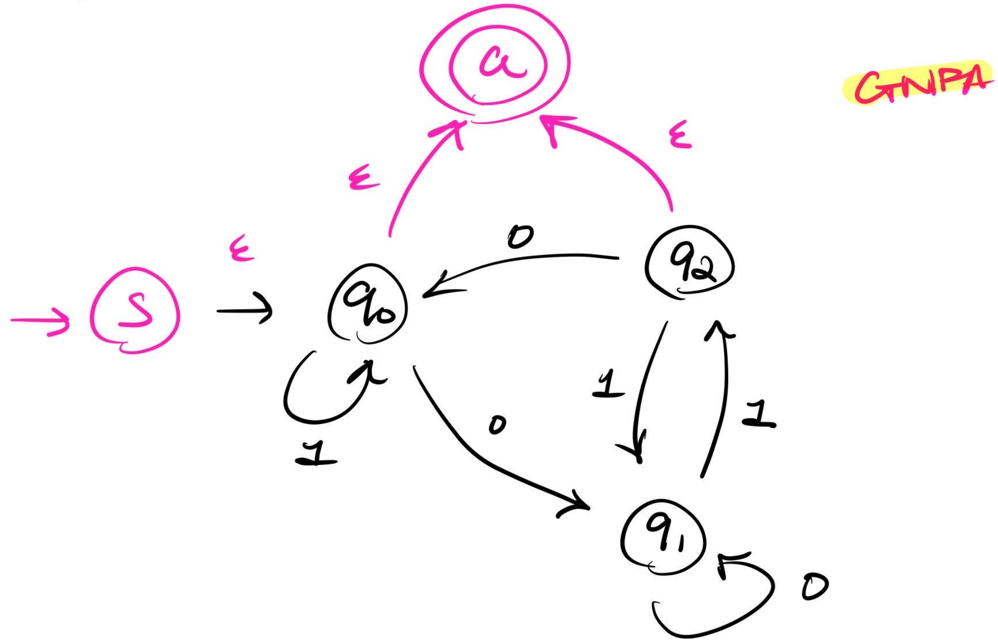
CS 3823 - Theory of Computation: Homework Assignment 2

5 DFAs to Regular Expressions [6 points]

Convert the following DFA to a regular expression.



Do so by starting from the equivalent GNFA, then remove the state q_1, then the state q_2 and finally the state q_0.



CS 3823 - Theory of Computation: Homework Assignment 2

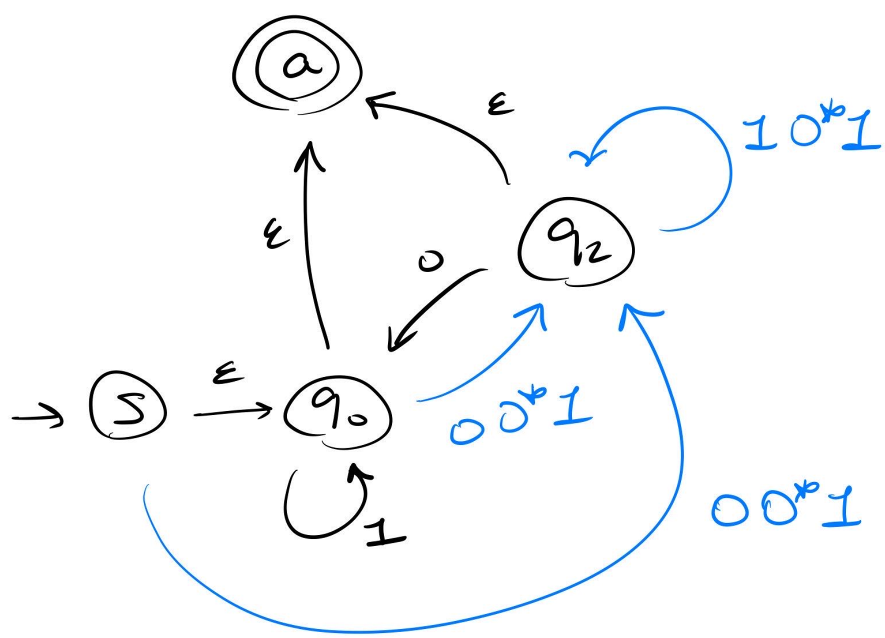
(blank space in case you need it for exercise 5)

(1) Remove q_1:

$q_0 \to q_2$:
$q_0 \to q_1 = 0$
$q_1 \to q_1 = 0^*$
$q_1 \to q_2 = 1$
$= 0\,0^*\,1$

$q_2 \to q_2$:
$q_2 \to q_1 = 1$
$q_1 \to q_1 = 0^*$
$q_1 \to q_2 = 1$
$= 1\,0^*\,1$

$s \to q_2$:
$s \to q_0 = \varepsilon$
$q_0 \to q_1 = 0$
$q_1 \to q_1 = 0^*$
$q_1 \to q_2 = 1$
$= \varepsilon 0 0^* 1 = 00^*1$



CS 3823 - Theory of Computation: Homework Assignment 2

(blank space in case you need it for exercise 5)

(2) Remove q_2 : $q_0 \to q_0$: $\to 1$

$\to (00^*1)(10^*1)^*0$

$= \underline{1 \cup (00^*1)(10^*1)^*0}$

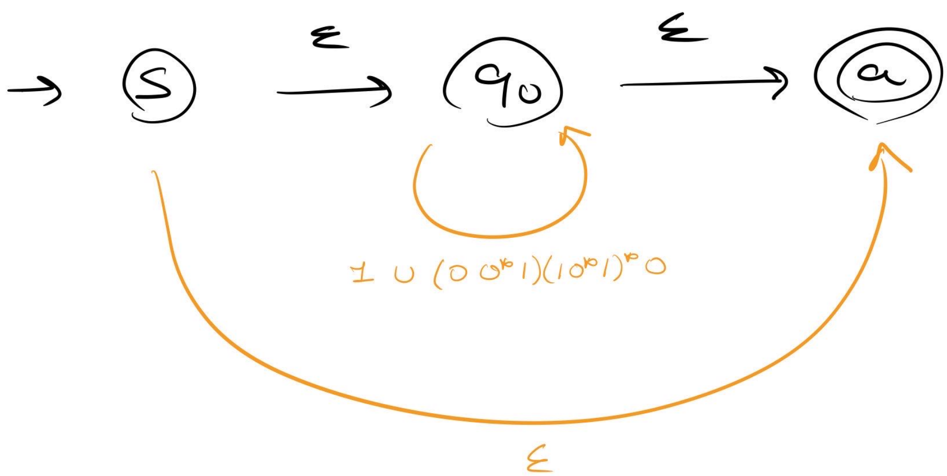
$s \to a$: $\to \varepsilon \cdot \varepsilon = \varepsilon$

$\to (\varepsilon)$

$\to (00^*1)(10^*1)\varepsilon$

$= \varepsilon \cup (00^*1)(10^*1)\varepsilon$

$= \underline{\varepsilon}$



CS 3823 - Theory of Computation: Homework Assignment 2

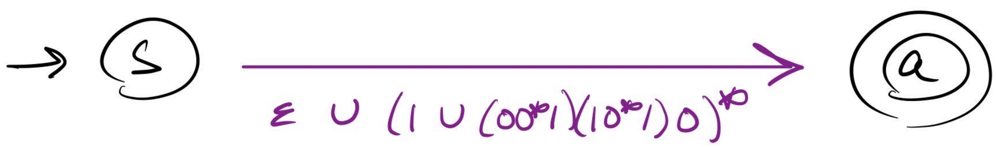
(blank space in case you need it for exercise 5)

(3) Remove q_0:

$s \to a$: $\to \varepsilon$

$\to (\varepsilon)(1 \cup (00^*1)(10^*1)^*0)^*(\varepsilon)$

$= \underline{\varepsilon \cup (1 \cup (00^*1)(10^*1)0)^*}$



6 Non Regular Languages [8 points]

Use the Pumping Lemma to prove that the following languages are not regular.

(i) $L_1 = \{www \mid w \in \Sigma^*\}$, $\Sigma = \{0, 1\}$.

Proof by contradiction: Start by assuming L_1 is regular, along with the following rules:

- *s* is any string in A of at least *p* length
- *s* can be divided into 3 pieces: xyz:
  - for i > 0,$xy^iz, z \in A$
  - $|y| > 0, |xy| \leqslant p$
  - $|xy| \leqslant p$

Let the commas denote where substrings x,y,z are delimited at:

$$s = 0101, 11, 011$$

Which holds true as our form is still in www. However, if we pumped y (11) to 111, we would now have

$$s = 101, 10, 1$$

it would then be invalid due to the original property not holding.

(ii) $L_2 = \left\{1^{2^n} \mid n \geqslant 1\right\}$, $\Sigma = \{1\}$.

Proof by contradiction: start by assuming L_2 is regular, along with the rules stated above in (i). Know length (s) of language must be some 2^n. Let the commas denote where substrings x,y,z are delimited at, and n=3:

$$s = 111, 11, 111$$

Which is a valid string for the language (length of s =8). If we pump the length of y to be longer, say xyyz we would end up with a new invalid set k:

$$k = 111, 1111, 111$$

Where k is length 10. Which shows that powers of 2 cannot be achieved by adding a fixed-length substring y. By inflating our y^i, the language does not fit the conditions to ensure the language is regular. From this k does not fit the language's property, the language must not be regular, QED