Computational Models – Exercise 3

Due Friday, 28 April 2023

Each student must solve the problems on their own. If you encounter difficulties, you may ask a classmate for a hint or the general idea. However, detailed discussion, note-taking, or sharing of written solutions is not allowed. Do not write down your answers while communicating with other people or show the answers for feedback.

Our grading app has severe limitations, such as no zoom tool. To make sure we can grade your work, please follow these technical guidelines:

Submit a **single PDF file** through Moodle.

The file size is limited to **10 MB**. If necessary, google *reduce PDF file size*.

Fill in your answers **on this form*** in the allocated spaces. The space provided gives you an indication of the expected length and level of detail of the answer. You may add a little more space if you need.

Include everything from this form in your submission. In particular, **include the problem statements**. Do not delete any text or omit pages, just add your answers.

Ensure your answers are **legible** (easy to read) at zoom 100% on a standard computer screen. Your text should be **large**, **sharp**, and in **high contrast** with the background.

Do not squeeze scanned solutions to fit in the space, as the text will become small.

Verify that pages are properly **ordered** and **oriented**.

The page size must be **A4**. Before submitting your file, check its page size using Acrobat Reader: go to File > Properties > Description and confirm that Page Size is around 21 × 29 cm. Note that scanning A4 pages does not guarantee the resulting page size will be A4, due to scaling. If necessary, google *resize PDF to A4*.

Do not add your answers as PDF comments. If you can drag them in Acrobat Reader, they are comments. If necessary, google *flatten PDF*.

A **5-point bonus** will be given to solutions typed in a word processor. Hand-sketched illustrations or diagrams will not deny you this bonus.

If there are technical issues with your submission, you may receive a fine. In extreme cases, your submission may not be graded at all.

If you need help or have questions, please use the course forum at Piazza.

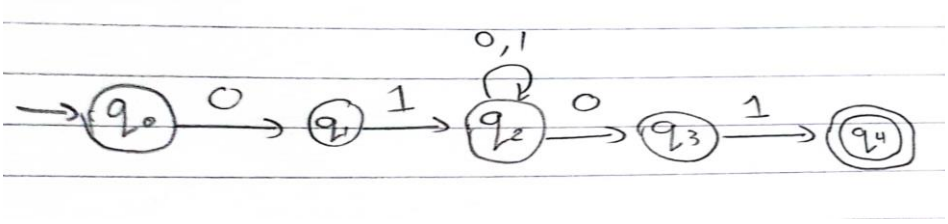
*The only exception is in case you use LaTeX or a similar typesetting system. In that case, copy-paste everything from this file, except for illustrations or other hard-to-reproduce graphical elements. No need to fix corrupted formulas.

Worked with Jemma Diamond - 806839

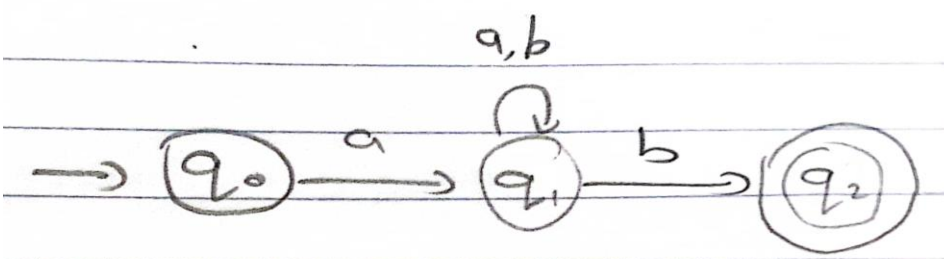
Problem 1

Define NFAs (non-deterministic finite automaton) for each of the following languages. It is enough to draw a state diagram for each language. Full credits will be given only for the NFA with the minimal possible states.

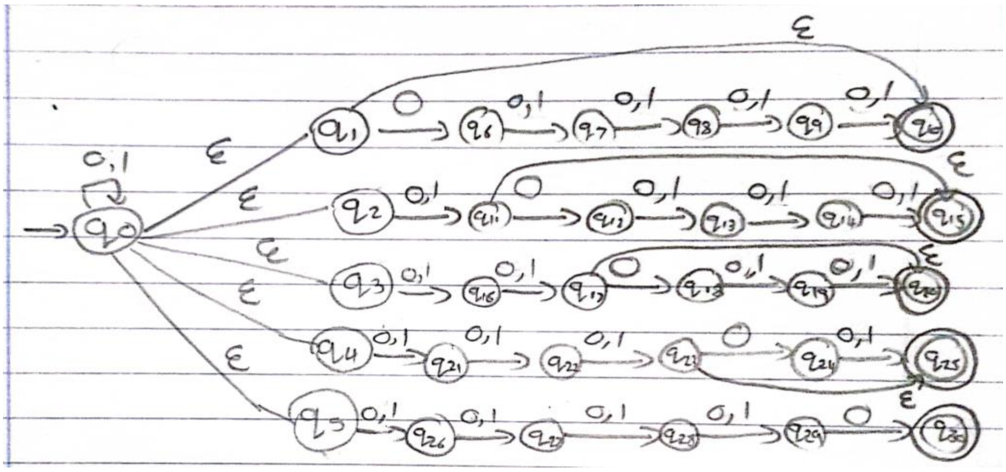
[4 pt] **1.** $L = \{w \in \{0,1\}^* | \, w \text{ starts with } 01 \text{ and ends with } 01\}$ (use maximum 6 states not including sink state).



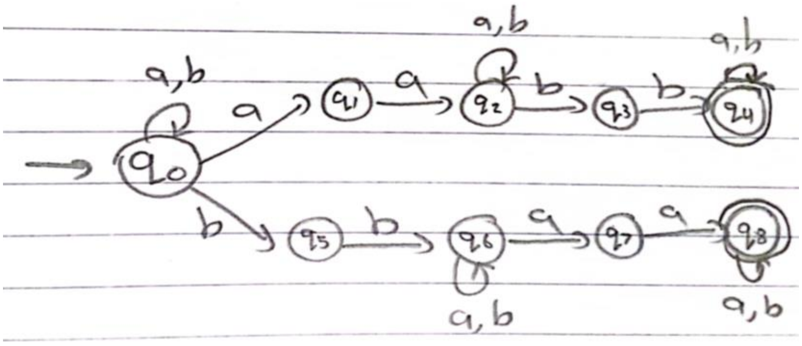
[4 pt] **2.** $L = \{a^n w b^n | \, n > 0, w \in \{a,b\}^* \}$ (use maximum 6 states not including sink state).



[4 pt] **3.** $L = \{w \in \{0,1\}^* | \text{ at least one of the last 5 letters is } 0\}$

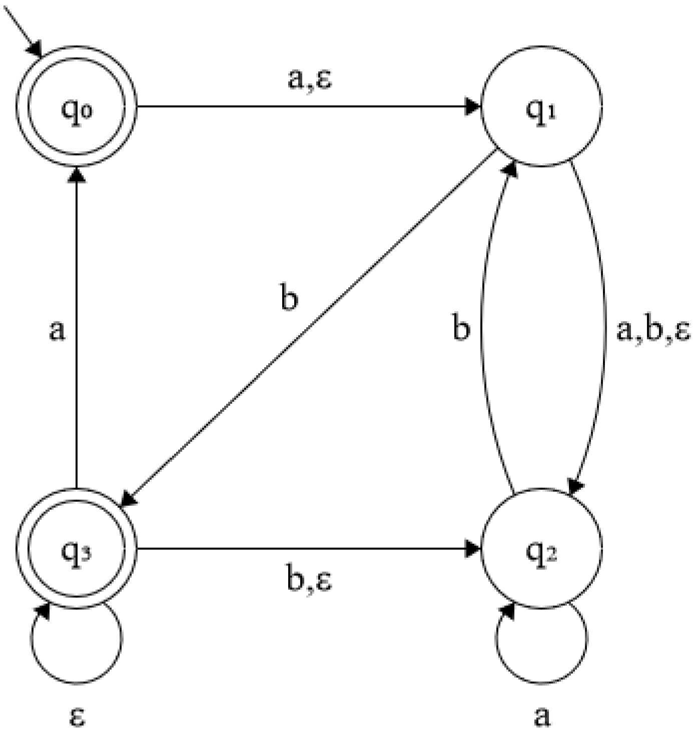


[4 pt] **4**. $L = \{w \in \{a,b\}^* | \; aa \text{ and } bb \text{ are substrings in } w\}$ (use maximum 9 states not including sink state).



[8 pt] **Problem 2**

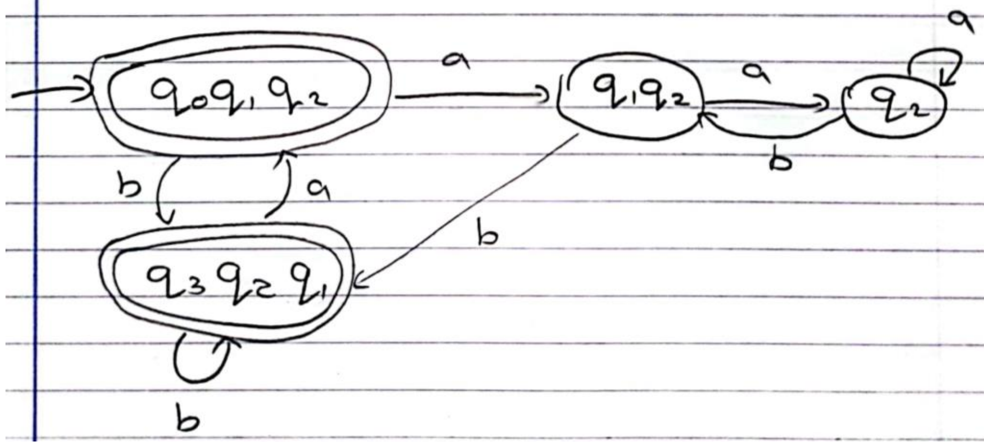
Define formally and draw an equivalent DFA for the following NFA. No need to prove your construction.



Define DFA, $D = \{Q_D, \Sigma, \delta_o, q_{sD}, F_D\}$ as:

$Q_D = P(Q_N)$
$\Sigma = \{a, b\}$
$q_{sD} = E(q_0) = \{q_0, q_1, q_2\}$
$F_D = \{R \in Q_D \mid q_0 \in R$ or $q_3 \in R\}$

δ_D	a	b
$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_3, q_2, q_1\}$
$\{q_1, q_2\}$	$\{q_2\}$	$\{q_3, q_2, q_1\}$
$\{q_3, q_2, q_1\}$	$\{q_0, q_1, q_2\}$	$\{q_2, q_1, q_3\}$
$\{q_2\}$	$\{q_2\}$	$\{q_1, q_2\}$



Problem 3

Prove by construction and/or regular closures:

[7 pt] **1**. For languages L_1 and L_2 we define: $L_1 \otimes L_2 = \{xyz \mid xz \in L_1 \; and \; y \notin L_2\}$.

Prove that if L_1 and L_2 are regular then $L_1 \otimes L_2$ is regular.

[7 pt] **2**. For a language L define $L' = \{w |\ ww^R \in L\}$. Prove that if L is regular, then L' is regular.

Problem 4

For each of the following languages, write a regular expression that is its language. If not stated specifically, the language is defined over $\Sigma = \{a, b\}$. A solution that is too long or complicated may not receive all the points.

[5 pt] **1.** $L = \{w \mid For\ each\ prefix\ u\ of\ w,\ |\#_a(u) - \#_b(u)| \leq 2\}$.

I don't know

[5 pt] **2.** $L = \left\{ \begin{matrix} All\ strings\ which\ start\ with\ \boldsymbol{a}\ have\ a\ substring \\ \boldsymbol{ab}\ and\ are\ of\ odd\ length \end{matrix} \right\}$.

L = a((a+b)(a+b))*ab((a+b)(a+b))* + a([(a+b)(a+b)]*(a+b))ab([[(a+b)(a+b)]*(a+b)) + ab([(a+b)(a+b)]*(a+b))

[5 pt] **3.** $L = \left\{ \begin{matrix} All\ strings\ which\ contain\ `\boldsymbol{abb}'\ as\ a\ subsring \\ and\ don't\ contain\ `\boldsymbol{aa}'\ as\ a\ substring \end{matrix} \right\}$.

L = (b* + (ab)*)abb(b* + (ab)*)

[5 pt] **4.** $L = \{a^n b^m \mid (n + m)\ mod\ 4 = 2\}$.

I don't know

Problem 5

Prove or disprove the following equivalence:

[5 pt] **1.** $a^*(b^*c^*)^* = a^*(b + c)^*$.

Denote r = a*(b*c*)* and denote s = a*(b + c)*.

Let w be part of L[s]= a*(b + c)*. If w= ε then w is part of L[r] by definition of *. Otherwise assume w has k instances of the letter a, therefore it can be written as w= $a_1a_2 ... a_k$w'. Assume w' has m instances of c, then w' can be written as w'= x_1c.... $x_m c x_{m+1}$ where each x_i is part of b*. Therefore w = $a_1a_2 ... a_k$w' = $a_1a_2 ... a_k$ (x_1c)...(x_mc) x_{m+1} = $a_1a_2 ... a_k$ (b*c)(b*c)...(b*c) = $a_1a_2 ... a_k$ (b*c*)* = a*(b*c*)*.

Let w be part of L[r]= a*(b*c*)*. if w= ε then w is part of L[s] by definition of *. Otherwise assume w has k instances of the letter a, therefore it can be written as w=$a_1a_2 ... a_k x_1 ... x_m$ where each x_i is part of (b*c*). Each $x_1 ... x_m$ can be seen as (b+c)*. All together w = $a_1a_2 ... a_k$ (b+c)* (b+c)*..... (b+c)* = a*(b+c)*.

[5 pt] **2.** $a^*(b + ab)^* = b + aa^*b^*$.

They are not equivalent. Counterexample:

Let w= ε. w is part of the language of a*(b + ab)* because of the definition of *, however w is not part of the language of b + aa*b* as this language only includes words that are "b", or words that start with an a, therefore it does not include the empty word.

[5 pt] **3.** $(ab^*)^*a = (a^*ba^*)^*a^*$.

They are not equivalent. Counterexample:

Let w= ε. w is part of the language of (a*ba*)*a* because of the definition of *, however w is not part of the language of (ab*)*a as this language includes words that must end with an "a" and therefore it does not include the empty word.

[5 pt] **4.** $a^*ba(bb)^* = ba(bba)^*$.

They are not equivalent. Counterexample:

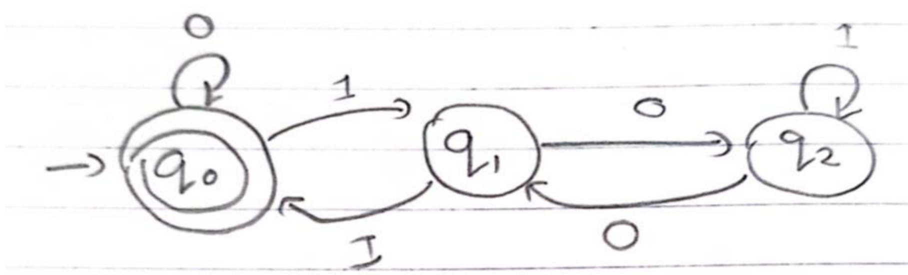
Let w= ababb. w is part of the language of $a^*ba(bb)^*$ (where a* = "a" and (bb)* = "bb"), however w is not part of the language of $ba(bba)^*$ as this language only includes words that start with an "ba" and therefore it does not include w = ababb.

Problem 6

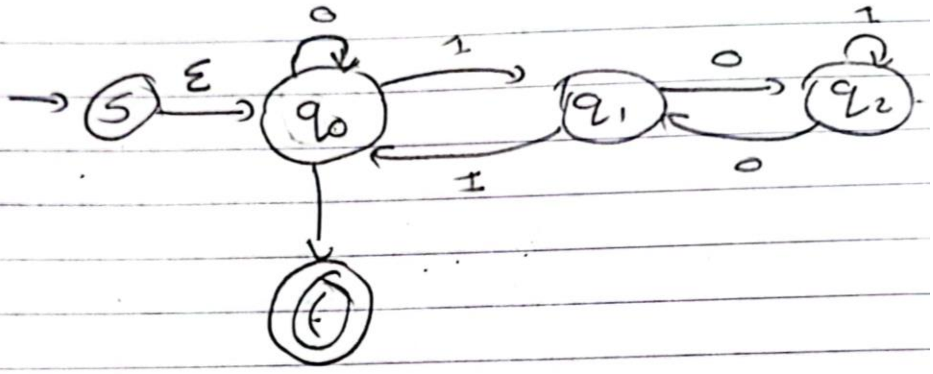
[8 pt] **1**. Build an NFA for the following regex (as we saw in class):
$0\big((0+1)(0+1)\big)^* + 1(0+1)\big((0+1)(0+1)\big)^*$.
Show the intermediate steps of your NFA construction.

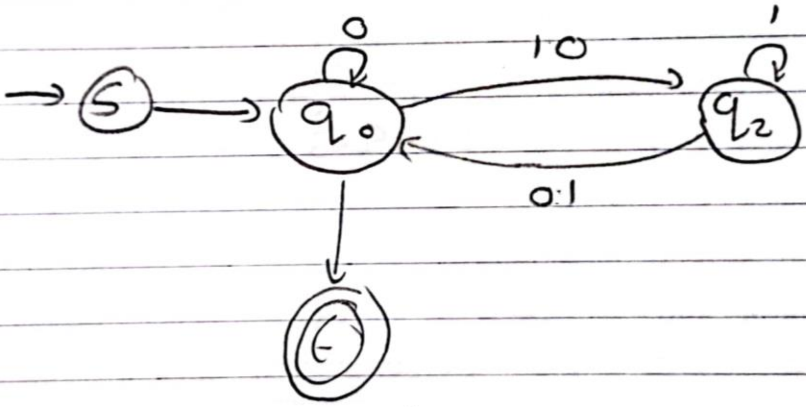
I don't know.

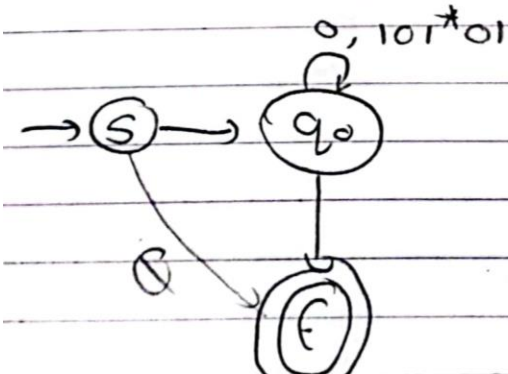
[5 pt] **2**. Build a DFA for the following language:
$L = \{w \in \{0,1\}^* | w \text{ is divisible by } 3 \text{ (can have a leading } 0's)\}$.
Hint: recitation 1.

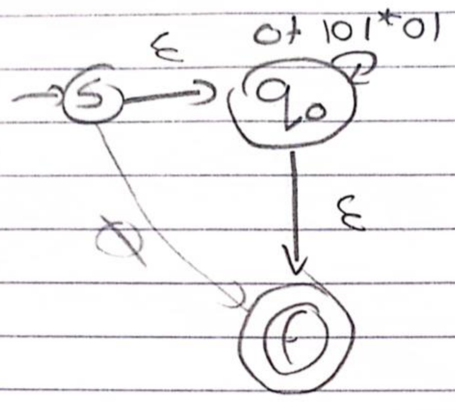


[9 pt] **3**. Write a regex for L (previous question) by converting your DFA into regex (as we saw in class). Show the intermediate steps of your regex construction.









$= \emptyset + \varepsilon(0+101^*01)^*\varepsilon$

$= (0 + 101^*01)^*$