CSE 3315

Theory of Computation

Fall 2024
Exam 1
Section 003

Last Name: Bhamal
First Name: Hasnain

Student ID: 1001937588

☑ If this box is checked (not required) I grant permission to Linda S. Barasch, PhD and Mary E. Koone to respectfully use my anonymized submissions for the purpose of teaching others.

By signing below, I certify that I will not receive help in taking this exam(from another student or from unauthorized materials), nor will I allow any student to receive help from me. I will not preserve by any means copies, images, or written record of any portion of this exam. I agree not to disclose the contents of this exam to anyone.

I understand that violation of this pledge is an ethics violation.

Signature Hasnain B

For all questions, you may explain your answer for possible partial credit.

$|s| > p$

$s = xyz$

(1) xy^iz $\quad i \geq 0$

(2) $p > 0$

(3) $|xy| < p$

(3) 1. The order of precedence for regular operations is * (star), concatenation (∘), ∪ (union).

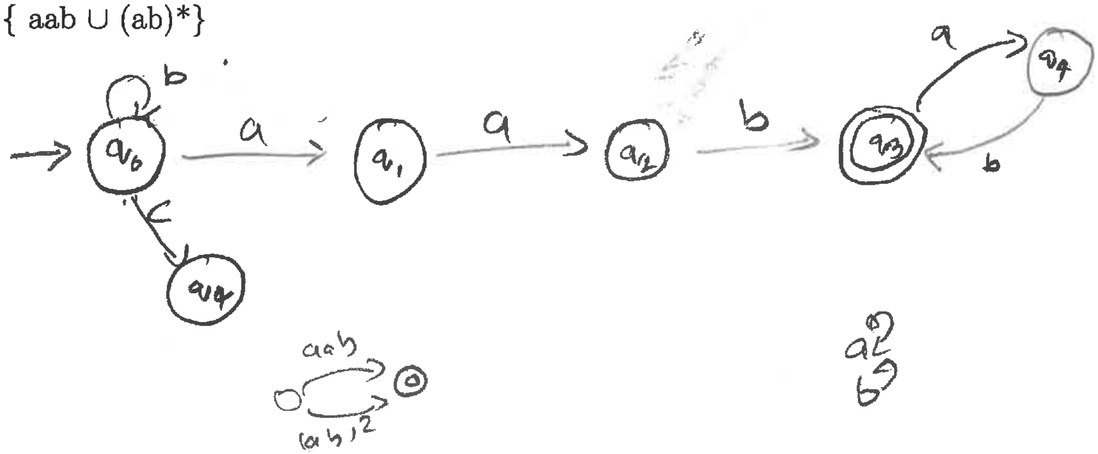
3

- ◉ True
- ○ False

(6) 2. (a) Build a DFA that recognizes language Z. *Let* $\Sigma = \{a, b, c\}$.

$Z = \{ aab \cup (ab)^* \}$

2



(5) (b) Give the full formal definition of language Z, from part (a) of this problem.

2

$Q = \{q_0, q_1, q_2, q_3, q_4\}$ -1

$\Sigma = \{a, b, c\}$

$\delta = Q \times \Sigma$ -1

$q_0 = q_0$

$F = q_3$ -1

(3) 3. Select ALL the true statements.

- ◯ A language is a set of strings.
- ◯ All sets of strings are regular languages.
- ◯ A string is a set of symbols.
- ◯ A string is a sequence of symbols.

2

(2) 4. List the elements of set A in shortlex order. A = {~~0~~, ~~11~~, ~~000~~10, ~~010~~, ~~10~~, ~~1~~, ~~00~~, ~~000~~, ~~ε~~}

2

A= {ε, 0, 1, 00, 10, 11, 000, 010, 00010}

(3) 5. How is the pigeonhole principle used in the proof of the pumping lemma for regular languages?

2

Pigeonhole principle states that for m number of items and n number of space, If m > n than there will be space where there is more than one item in the space. Pumping lemma uses the same principle by pumping the string, and stating if the language is regular or not. Even that's the reason why we consider pumping lemma as a negative proof coz stating the long is regular by pumping lemmas has only some confidence but proving its not, is 100% sure

(5) 6. Select the items that are required for a Nondeterministic Finite Automaton (NFA) to accept a string.
Select all that apply.

2

- ◯ At least one empty transition must be followed.
- ◯ There exists a path that ends on a final (accept) state.
- ◯ All states must be reachable.
- ◯ The entire input is consumed. −1
- ◯ Operation proceeds according to the transition function. −1
- ◯ The input string is at least as long as the pumping length.
- ◯ Operation begins at the start state.
- ◯ All paths must end on a final (accept) state. −1

(3) 7. Informally describe the SUBSET-SUM problem.
A formal description is acceptable but not required.

3

in a subset sum problem, there is a sum already given. and if sum of a subset of a set is equal to the sum already given it is called a SUBSET-SUM problem

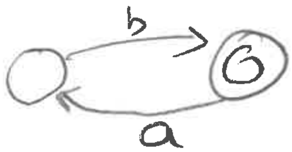
(8) 8. Using NFA closure and the inductive definition of regular expressions, as discussed in class, build an NFA equivalent to this Regular Expression.

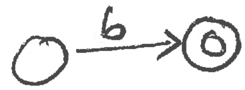
3.5

Remember to show the empty transitions.

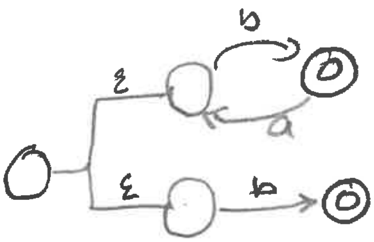
$((ba)^* \cup b)a^*$

$(ba)^*$

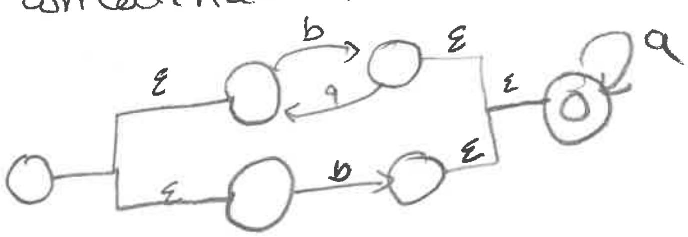




(2) union same start state



(2) concatenation



(6) 9. Using definitions and theorems discussed in class, prove the following.

A language L is regular if and only if (*iff*) some regular expression describes (generates) it.

0

A language is regular can be verified in 4 ways

(1) has a DFA

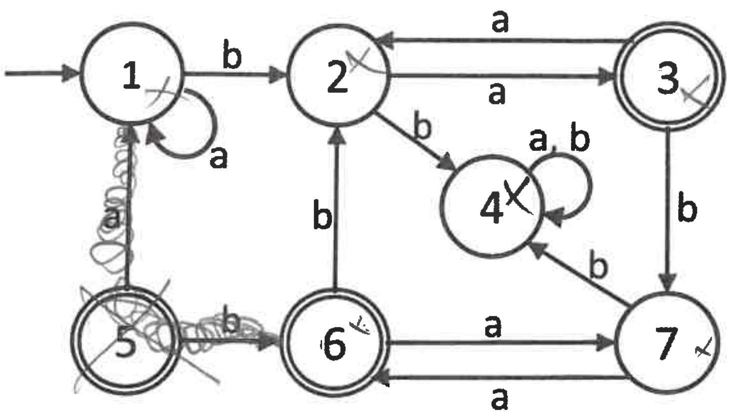
(2) has a NFA

(3) is in Regular Language

(4) performs closure like, Union, concatenation & star

(8) 10. Perform the state minimization algorithm *as given in class* on this DFA and sketch the state diagram of the resulting minimized DFA.

If the DFA is already in minimum form, no additional state diagram is needed.



Pseudo code

(1) Mark state
(2) for unmarked state follow the incoming transition, and if its from a marked state, mark that state too are ob
(3) eliminate it

step 1: eliminate 5
step 2: perform the table & merge similar

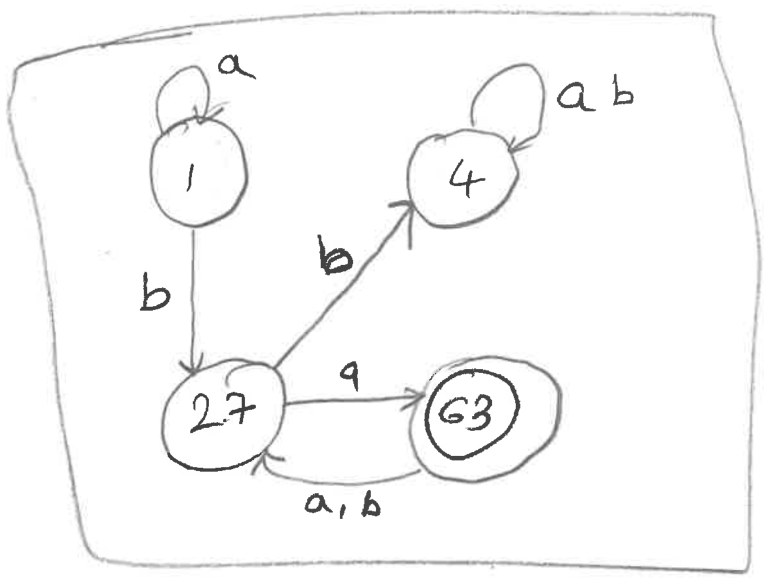
	a	b
1	1	2
- 2	3	4
4	4	4
- 7	6	4

	a	b
6	7	2
3	2	7

1 2 4 7   6 3

14 27   63

1 4 27 63



(10) 11. Using the pumping lemma, prove that L is not regular. Use the closure properties of regular languages if appropriate.

$L = \{a^n b^m a^{2n} | n, m \geq 0\}$

In other words, strings in this language have some quantity the symbol a, followed by some quantity of the symbol b, followed by twice as many a symbols as there were in the first part of the string.

-1 Assume L reg

So we are given some string $s = L$, where some quantity of symbol b, followed by twice as many a symbol as there were in first part of string

Pumping lemma can be used to prove if it is regular or not for that lets assume some amt of length p, where $|s| > p$ and $s = xyz$, for that we have:

(i) $s = xy^iz$, where $i \geq 0$ & $s \in L$
(ii) $p > 0$ -1
(iii) $|xy| \leq p$

to prove that our $s = a^n b^m a^{2n}$
$= a^p b^p a^{2p}$ here we can assume $m = p$ since, there is no query given for it & $m, n \geq 0$

lets say our $y = a^p$

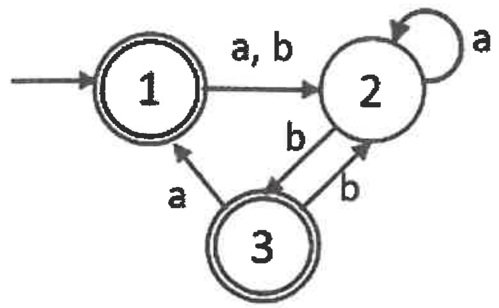
lets pump our s with 2, which will give us equal amount of a's in the first and last, and by doing that we can see it'll not be in the language it should be in language, coz by condition (i) its not in the language hence more a's, Hence

-2

language

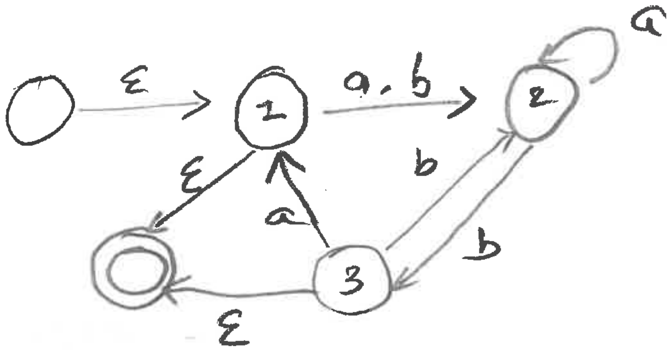
(10) 12. Using the DFA to Regular Expression algorithm discussed in class, find a regular expression for this DFA. Label each step. Sketch the resulting DFA after each major step.

Handle states in the order they are numbered.

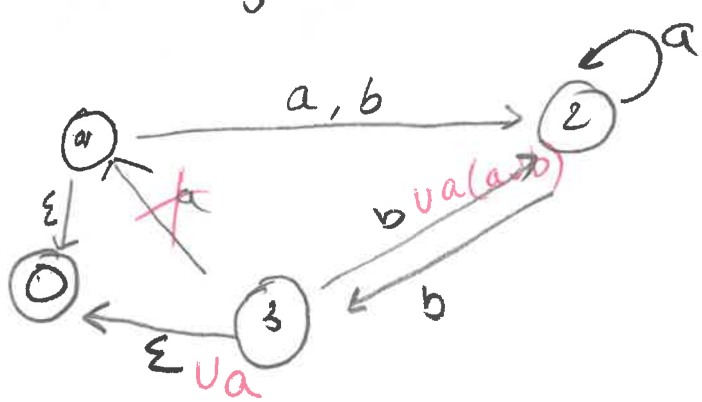


Step - 1 : make new start & finish state

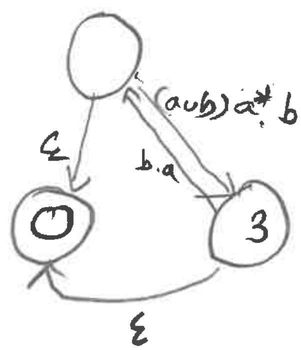
step - 2 elimina 1

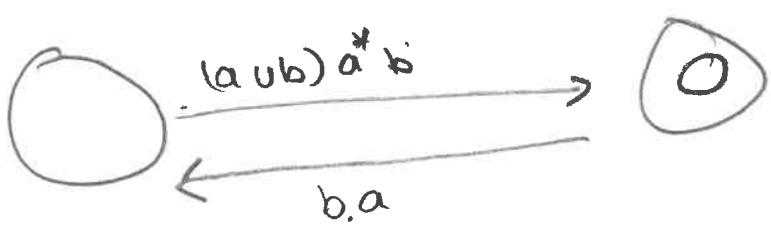


→ Eliminating 1



Eliminating ②

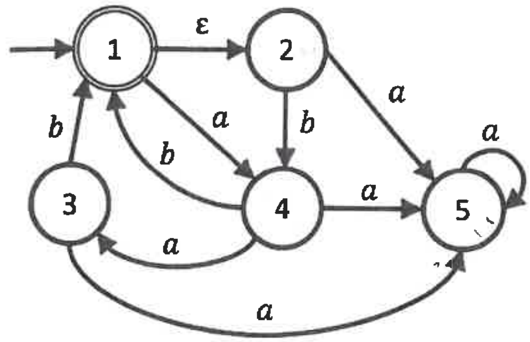




$\Rightarrow \{((a \cup b)a^* \cancel{b)} \cup ba\}$

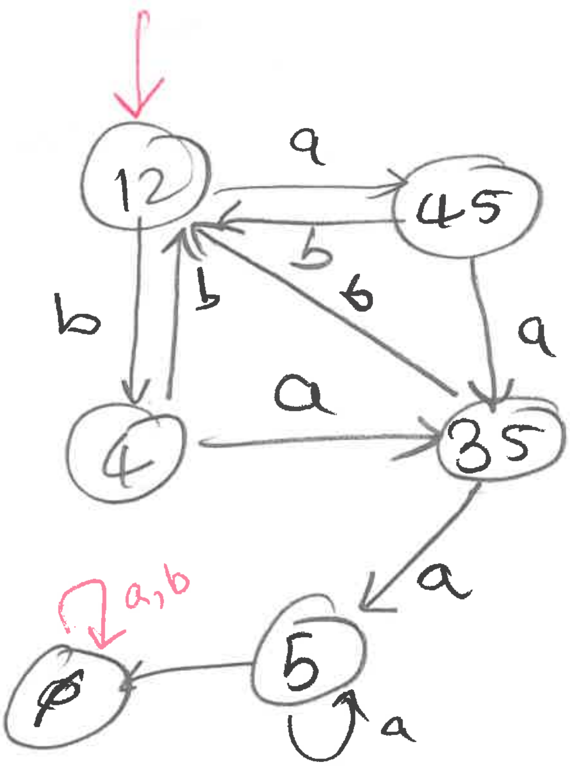
(10) 13. Using the algorithm discussed in class, (NFA → DFA) build a DFA that is equivalent to this NFA.

The algorithm does not include minimization, so do not reduce the number of states produced.



	a	b
1	425	24
425	35	12
24	5	41
35	5	1
[illegible]	45	

12



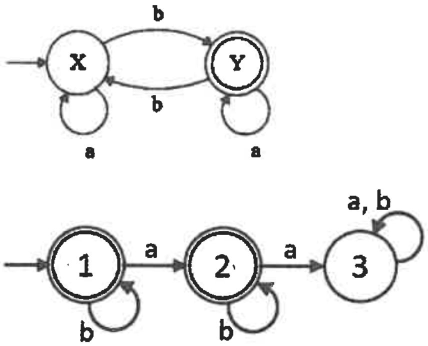
Final states?
-2

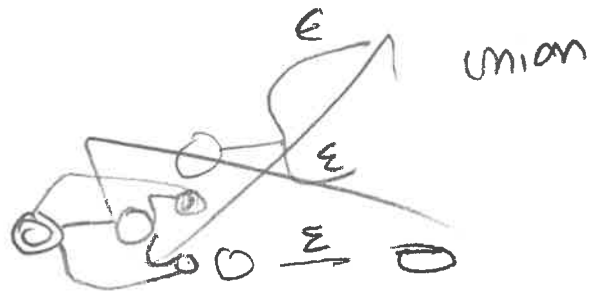
	a	b
12	45	4
45	35	12
4	35	12
35	5	12
5	5	∅

(4) 14. (a) For **each** DFA, write a succinct, equivalent regular expression.

The DFA → RE algorithm might not give a succinct RE and is not required here.

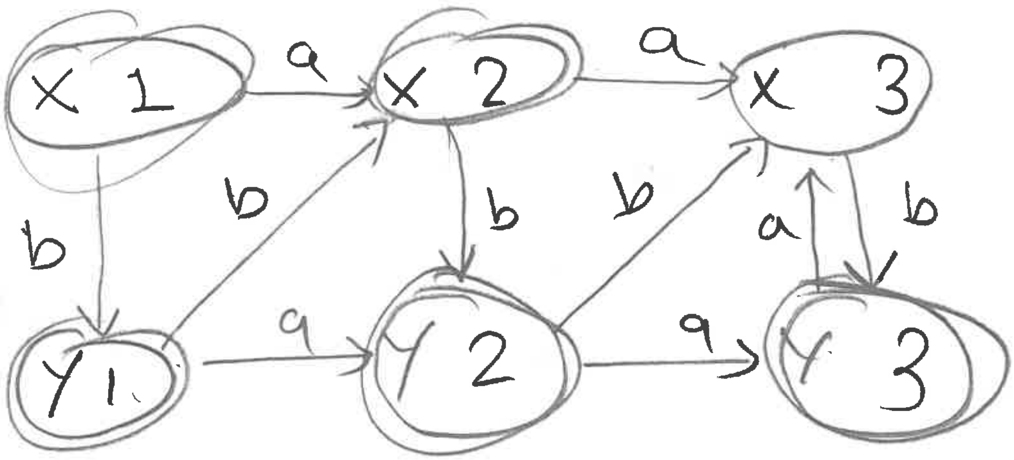
0





(6) (b) Using the algorithm discussed in class, "cross product," build a DFA that recognizes the <u>intersection</u> (∩) of the languages recognized by the machines above.

0



(2) (c) What would the set of final (accepting) states be so this DFA recognizes the **union** of the original two languages?

2

{ x1, x2, y1, y2, y3 }