CS 301 Homework 2

Ryan Magdaleno

TOTAL POINTS

30 / 40

QUESTION 1

1 NFA to DFA **7 / 10**

- **- 0 pts** Correct

Valid DFA

- **- 1 pts** No start state
- **- 1 pts** No final state
- **- 2 pts** One state transition for each character

✓ **- 2 pts** *Epsilon incorrect*

- **- 1 pts** Incorrect final states

Transitions

- **- 3 pts** No transitions

✓ **- 1 pts** *One incorrect transition*

- **- 2 pts** Two or more incorrect transitions

- **- 10 pts** No submission

1 A on 1 goes to {C,D,A}

QUESTION 2

Regex to NFA 20 pts

2.1 (abc)* U (ab)* **10 / 10**

Valid NFA

✓ **+ 1 pts** *Start state*

✓ **+ 1 pts** *At least 1 final state*

✓ **+ 1 pts** *Has transitions*

- **+ 0 pts** Click here to replace this description.

✓ **+ 3 pts** *Accepts empty string*

✓ **+ 2 pts** *Accepts 'abc' and 'ab'*

✓ **+ 2 pts** *Accepts $$(abc)^x$$ and $$(ab)^x$$*

- **+ 0 pts** No submission
- **- 2 pts** accepts 'ababc'
- **- 2 pts** accepts (ab)*(abc)* or (abc)*(ab)*

2.2 ((ab)*c)* **10 / 10**

Valid NFA

✓ **+ 1 pts** *Start state*

✓ **+ 1 pts** *At least 1 final state*

✓ **+ 1 pts** *Has transitions*

✓ **+ 1 pts** *Accepts the empty string*

✓ **+ 2 pts** *Does **not** accept 'ab'*

✓ **+ 2 pts** *Accepts 'abc'*

✓ **+ 2 pts** *Accepts $$c^*$$*

- **- 1 pts** Missing strings of the format such as $$c^n(abc)^m$$
- **+ 0 pts** No submission

QUESTION 3

3 NFA to Regex **3 / 10**

✓ **+ 2 pts** *Valid regex*

First removal GNFA (B or C)

- **+ 4 pts** All transitions ok
- **+ 3 pts** 3 transitions ok
- **+ 2 pts** 2 transitions ok

✓ **+ 1 pts** *1 transition ok*

Final regex

+ 1 pts Outer [00*1 ∪] or [11*0 ∪]

+ 1 pts Inner [1∪00*1] or [0∪11*0]

+ 1 pts Inner middle Kleene star

+ 1 pts Inner [0∪00*1] or [1∪11*0]

+ 0 pts No submission

2 A to D is 00*1

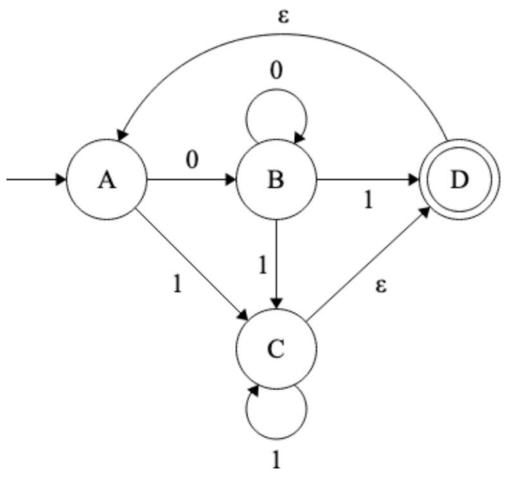
Homework 2

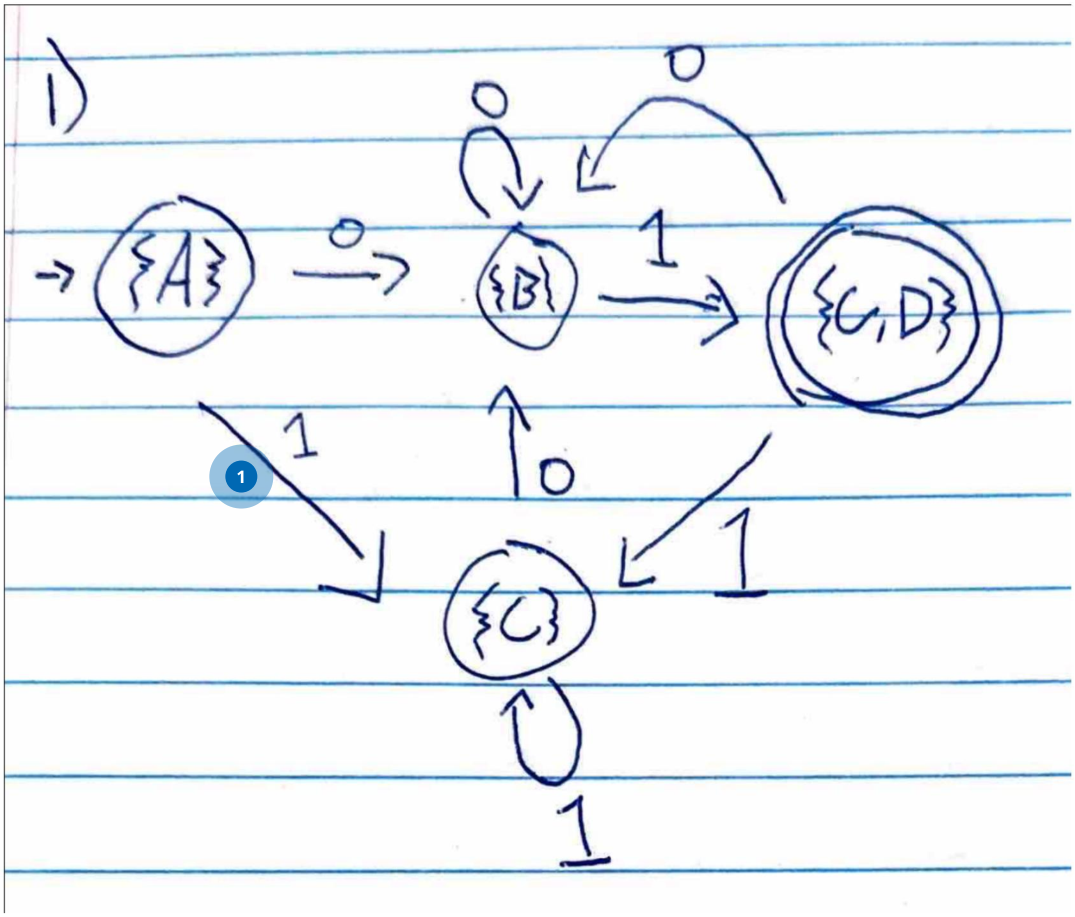
Due: September 20th, 8:00pm on Gradescope

All work must be individual

1 NFA to DFA

Provide the state diagram for a DFA that accepts the same language as the NFA M below. Make sure you have a start state and final state(s) labeled as well as exactly one transition for *each* state, character pair. You do **not** need to provide the 5-tuple.





1 NFA to DFA 7 / 10

- **0 pts** Correct

Valid DFA

- **1 pts** No start state
- **1 pts** No final state
- **2 pts** One state transition for each character

✓ **- 2 pts** *Epsilon incorrect*

- **1 pts** Incorrect final states

Transitions

- **3 pts** No transitions

✓ **- 1 pts** *One incorrect transition*

- **2 pts** Two or more incorrect transitions

- **10 pts** No submission

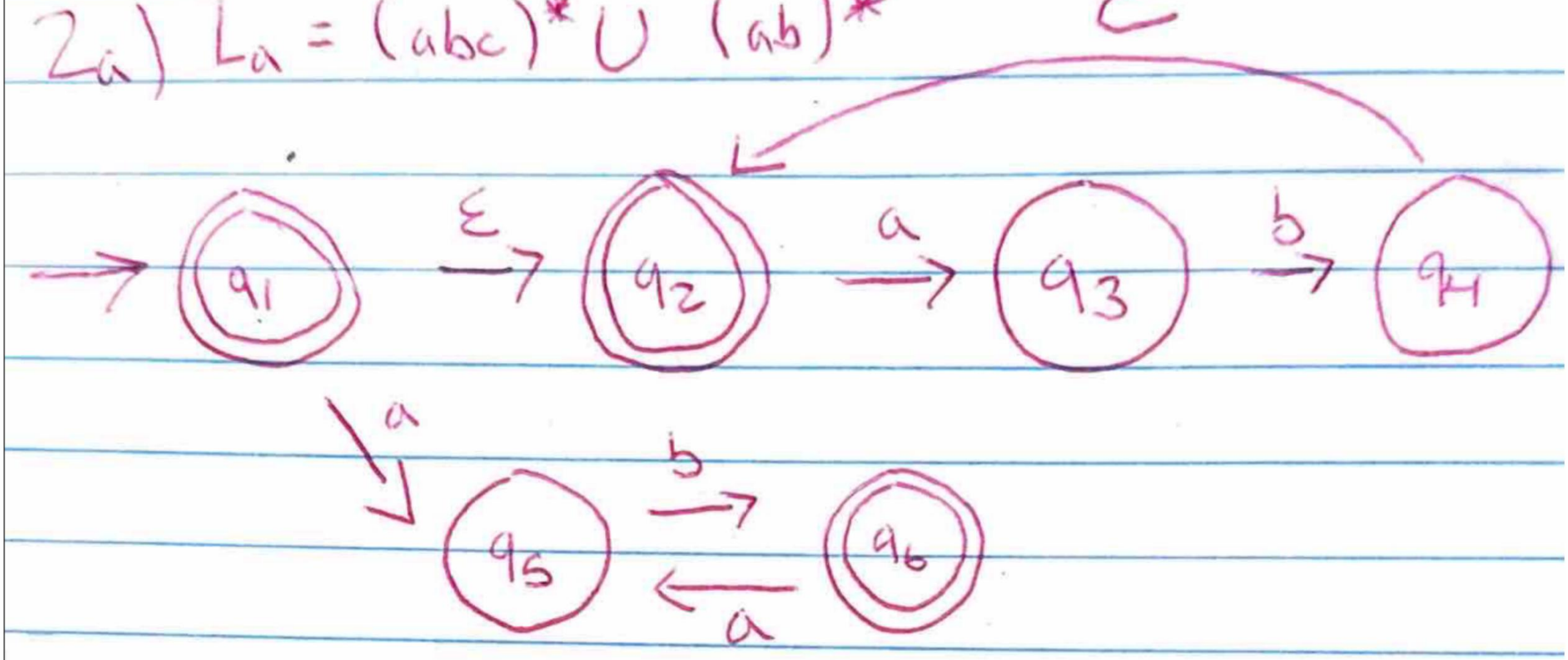
● 1 A on 1 goes to {C,D,A}

2 Regular Expression to NFA

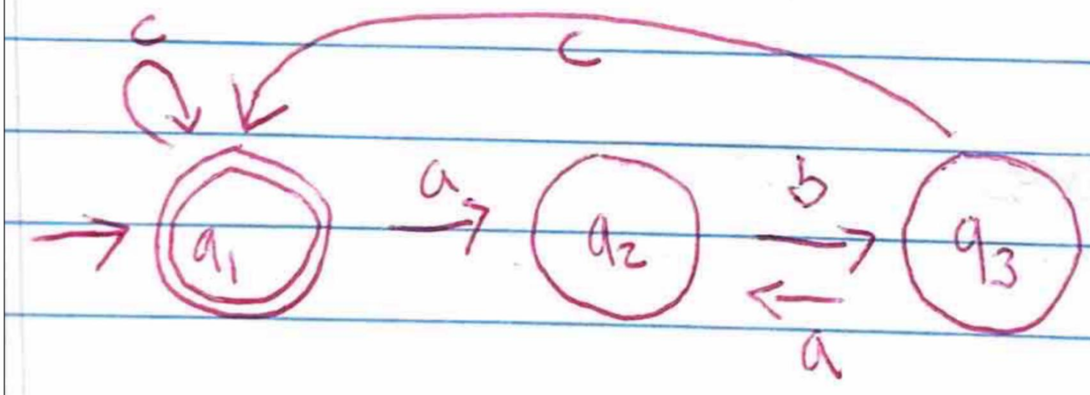
Produce the state diagram for a NFA which decides the following languages. $\Sigma = \{a, b, c\}$. *You do not need to produce the 5-tuple.*

a) $L_a = (abc)^* \cup (ab)^*$

2a) $L_a = (abc)^* \cup (ab)^*$



2b) $L_b = ((ab)^* c)^*$



2.1 (abc)* U (ab)* **10 / 10**

Valid NFA

✓ **+ 1 pts** *Start state*

✓ **+ 1 pts** *At least 1 final state*

✓ **+ 1 pts** *Has transitions*

+ 0 pts Click here to replace this description.

✓ **+ 3 pts** *Accepts empty string*

✓ **+ 2 pts** *Accepts 'abc' and 'ab'*

✓ **+ 2 pts** *Accepts $$(abc)^x$$ and $$(ab)^x$$*

+ 0 pts No submission

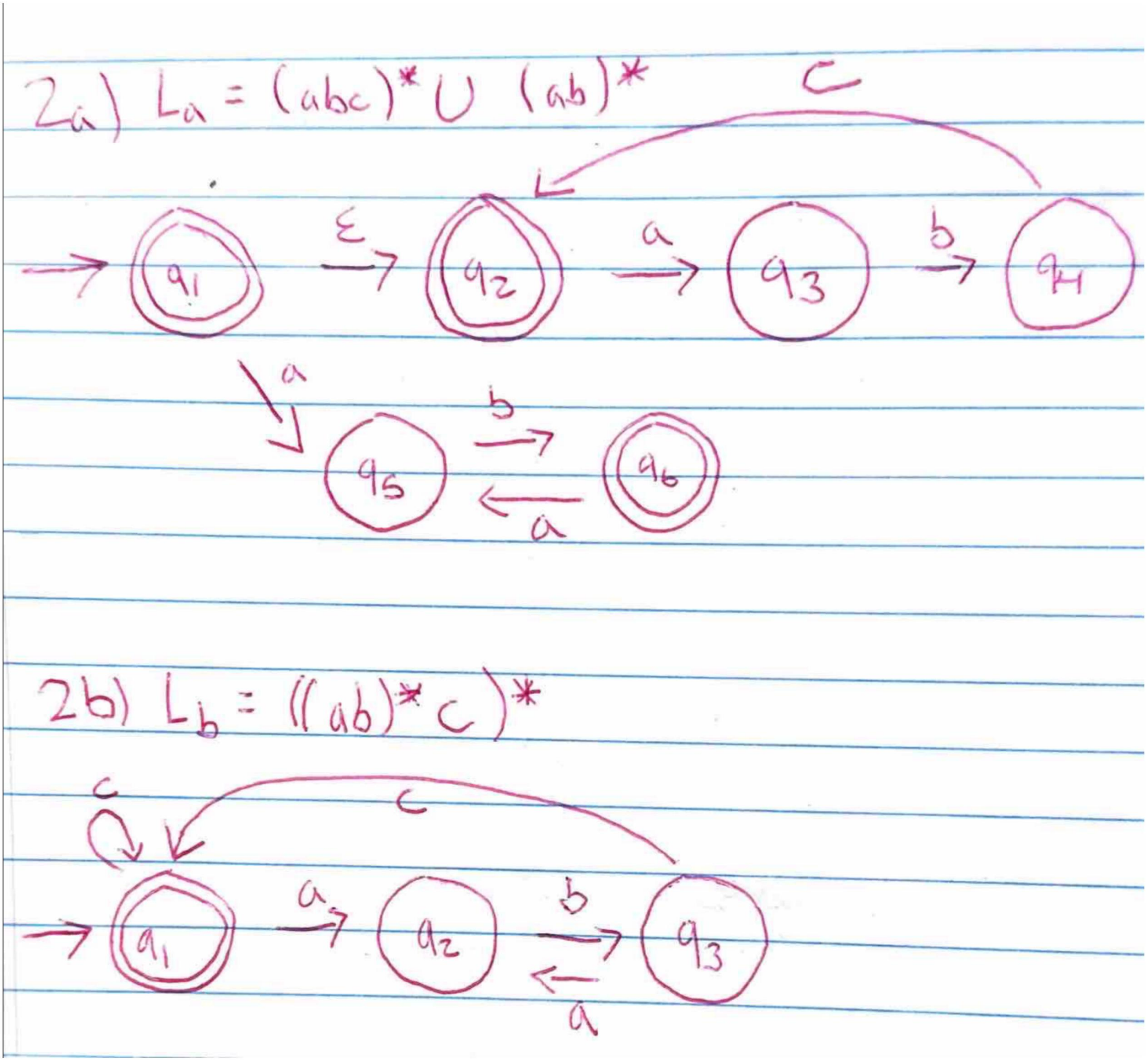
- 2 pts accepts 'ababc'

- 2 pts accepts (ab)*(abc)* or (abc)*(ab)*

2 Regular Expression to NFA

Produce the state diagram for a NFA which decides the following languages. $\Sigma = \{a, b, c\}$. *You do not need to produce the 5-tuple.*

a) $L_a = (abc)^* \cup (ab)^*$



2a) $L_a = (abc)^* \cup (ab)^*$

2b) $L_b = ((ab)^* c)^*$

2.2 ((ab)*c)* 10 / 10

Valid NFA

✓ **+ 1 pts** *Start state*

✓ **+ 1 pts** *At least 1 final state*

✓ **+ 1 pts** *Has transitions*

✓ **+ 1 pts** *Accepts the empty string*

✓ **+ 2 pts** *Does **not** accept 'ab'*

✓ **+ 2 pts** *Accepts 'abc'*

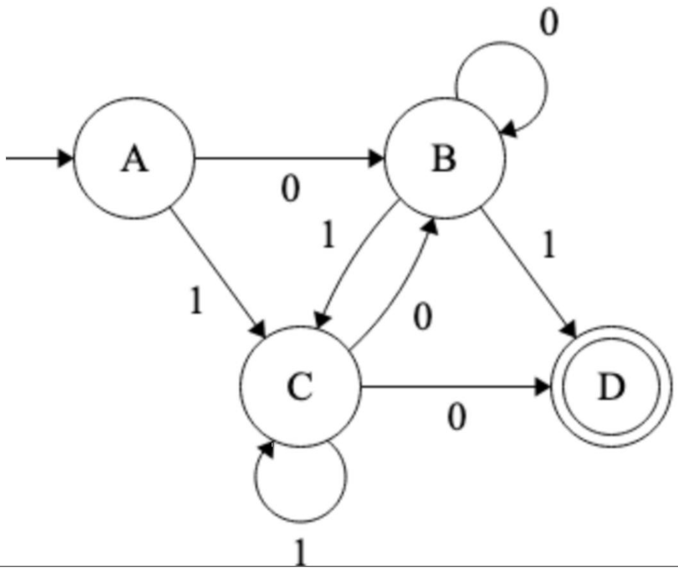
✓ **+ 2 pts** *Accepts $$c^*$$*

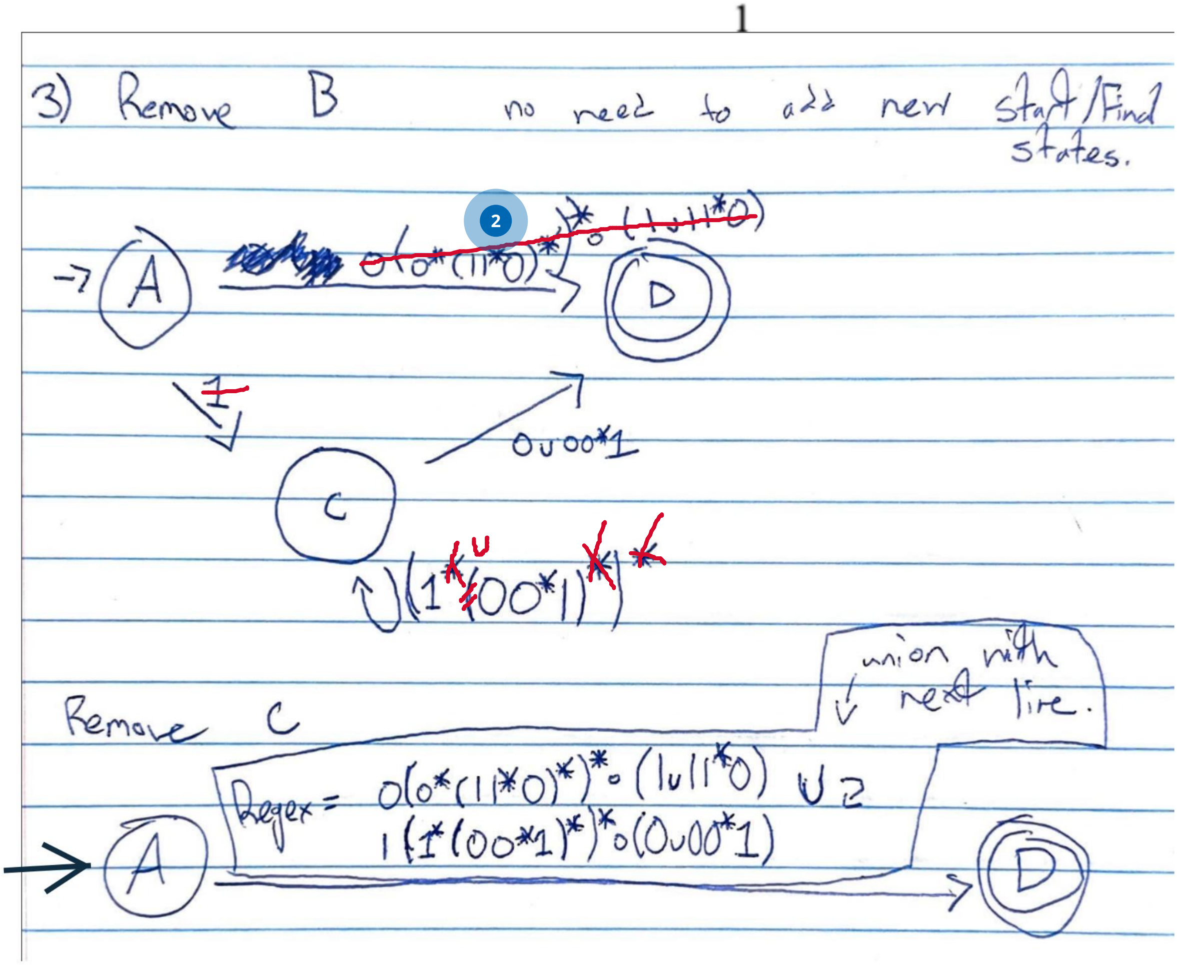
- 1 pts Missing strings of the format such as $$c^n(abc)^m$$

+ 0 pts No submission

3 NFA to Regular Expressions

Give a regular expression for the language decided by the following NFA M. Let $\Sigma = \{0, 1\}$ Show the intermediate GNFAs after removing each state.





3) Remove B

no need to add new start/Final states.

Remove C

union with next line.

Regex = $0(0^*(11^*0)^*)^* \circ (1 \cup 11^*0) \cup 2$
$1(1^*(00^*1)^*)^* \circ (0 \cup 00^*1)$

PCRE2 Regex:
^(0(0*(11*0)*)*(1|11*0)|1(1*(00*1)*)*(0|00*1))$

https://regex101.com/

3 NFA to Regex 3 / 10

✓ **+ 2 pts** *Valid regex*

First removal GNFA (B or C)

- **+ 4 pts** All transitions ok
- **+ 3 pts** 3 transitions ok
- **+ 2 pts** 2 transitions ok
- ✓ **+ 1 pts** *1 transition ok*

Final regex

- **+ 1 pts** Outer [00*1 ∪] or [11*0 ∪]
- **+ 1 pts** Inner [1∪00*1] or [0∪11*0]
- **+ 1 pts** Inner middle Kleene star
- **+ 1 pts** Inner [0∪00*1] or [1∪11*0]

- **+ 0 pts** No submission

2 A to D is 00*1