CS 228: Logic in CS	Autumn 2024

Quiz 2

Total Marks: 20 — *16 October 2024*

1. **[4 marks]** Let L be a non-empty regular language over an alphabet Σ. We denote by

   - $\mathsf{short}(L)$ the length of the shortest word in L.
   - $\mathsf{det}(L)$ the number of states of the smallest DFA A that accepts L.
   - $\mathsf{nd}(L)$ the minimal number of states in any NFA that accepts L.

   Suppose $|\Sigma| = 1$. Which of the statements below are true?

   (a) For every integer $k \geq 1$, there exists regular language L_k such that $\mathsf{short}(L_k) = \mathsf{nd}(L_k)$ but $\mathsf{nd}(L_k) \neq \mathsf{det}(L_k)$.

   (b) For every integer $k \geq 1$, there exists regular language L_k such that $\mathsf{short}(L_k) = \mathsf{nd}(L_k) - 1$ but $\mathsf{nd}(L_k) = \mathsf{det}(L_k)$.

   (c) For every integer $k \geq 1$, there exists regular language L_k such that $\mathsf{short}(L_k) < \mathsf{nd}(L_k)$ but $\mathsf{nd}(L_k) = \mathsf{det}(L_k) + 1$.

   Give proper explanation to your answer. Answers without explanation are bound to lose marks.

   Solution

   (a) and (c) are false, and (b) is true.

   Consider the language L_k given by the regular expression $a^{k-1}a^*$. Then, for any $k > 1$, $\mathsf{short}(L) = k - 1$. Further $\mathsf{nd}(L) = \mathsf{det}(L) = k$. Thus, (b) is true. $\mathsf{short}(L) < \mathsf{nd}(L)$ for any L so (a) is false, Finally, $\mathsf{nd}(L) \leq \mathsf{det}(L)$ for any L so (c) is also false.

   Rubrics

   - For (a), 1 mark has been given if and only if $\mathsf{short}(L) \leq \mathsf{nd}(L) - 1$ was obtained.
   - For (b), 1 mark has been given for specifying the language which satisfies the property (for eg $a^{k-1}a^*$) and 1 mark for the proof why this language satisfies both properties.
   - For (c), if you have stated that a DFA is also an NFA or that nd(L) $\leq$ det(L), you have been given 1 mark. If it isn't clear or that, there is something wrong written along with this then 0.5 marks have been given instead.

2. **[10 marks]** Consider the sentence $\varphi = \forall x(Q_a(x) \rightarrow \exists y[S(x, y) \wedge Q_b(y)])$. Draw the DFA A for which $L(A) = L(\varphi)$ **exhibiting all the steps** as discussed in class.

Solution

The given formula is $\varphi = \forall x(Q_a(x) \to \exists y[S(x,y) \wedge Q_b(y)])$.

We will work with $\varphi = \neg\,\exists x\,(Q_a(x) \wedge \neg\,(\exists y\,[S(x,y) \wedge Q_b(y)]))$.
All details of the steps involved in the construction of the automation can be found in the Appendix at the end. The marks for each step is also indicated.

Rubrics

Notice that there is no unique procedure involved in the construction. Any valid **procedure** is accepted. However, as the question demands you are expected to show all steps of construction. A *possible* allocation of marks at each step is as shown in the Appendix, along with detailed solution.

- What we have shown is the minimal DFA (the one with the smallest number of states after pruning out unnecessary *trap* states). We do not expect a *minimal* automata, any correct solution with any number of states are given full credit.
- More importance is given to the **procedure**. So explicitly showing each steps is important. However, we haven't penalised skipping trivial, or easy steps. For example, drawing the automaton for $S(x,y) \wedge Q_b(y)$ directly is accepted, and in that case we have included the marks of $S(x,y)$ and $Q_b(y)$ along with the automaton for $S(x,y) \wedge Q_b(y)$. We have also given marks even if the automaton of $Q_a(x)$ is not shown explicitly. Similarly skipping small steps in between is also taken care of.
- However, we *require* you to show steps involving *intersection*, *complementation*, *determinization* of automata, as they are very important. We have deducted marks for skipping such steps.
- Since we focus on the **procedure** involved, if you have made any mistake in between, we have not considered the mistake in future steps. What we have looked into is whether you know how to do *intersection*, *complementation*, *determinization*, etc. So, in case of mistakes in a step, we have deducted marks at that step and given marks considering that you are consistent with the mistake ahead (with some penalty).
- Some marks is dedicated to the final DFA. However, if only the final DFA is drawn, atmost 2 marks is given.
- In the question nothing was explicitly mentioned about the alphabet Σ. Specifically, it was not mentioned that $\Sigma = \{a, b\}$. So you were supposed to work with any $\Sigma \supseteq \{a, b\}$. However, we have not deducted any marks for using $\Sigma = \{a, b\}$.

IMPORTANT: We have graded this question in the most liberal manner. Ensure that you raise only valid cribs. Valid cribs include counting mistakes and marks not given for a step as mentioned in rubrics. Unnecessary arguments and cribs will attract a penalty of **-2 marks**.

Suggestions for Future Exams. The following are some minor mistakes we noticed. Be sure to avoid them in future. We have been too lenient in giving no penalty or minimal penalty in such cases.

- Make sure you clearly indicate the start states of automata. Next time, we will not consider any automaton without indicating start state as valid.
- The final states of automata determine its language. Make sure you **clearly** indicate which are final states and which are not. Unclear or confusing marking of final states will be strictly penalised next time.
- Show the transitions clearly and neatly.
- Whenever you construct a DFA make sure to show all transitions. Otherwise, it will be counted as an NFA.

3. **[6 marks]** Let Σ, Γ be two finite alphabets. Define a function $\mathsf{en} : \Sigma \mapsto \Gamma^*$ which defines an encoding of symbols of Σ as words over Γ^* such that $\mathsf{en}(ab) = \mathsf{en}(a).\mathsf{en}(b)$ for any $a, b \in \Sigma$. For example, consider $\Sigma = \{a, b\}, \Gamma = \{c, d\}$ and define $\mathsf{en}(a) = \epsilon, \mathsf{en}(b) = cdc$. Then $\mathsf{en}(aabb) = cdccdc$.

   For a language $L \subseteq \Sigma^*$, define
   $$\mathsf{en}(L) = \{\mathsf{en}(w) \mid w \in L\}$$
   Prove or disprove : If L is FO-definable then $\mathsf{en}(\mathsf{L})$ is MSO-definable.

   If you think the statement is true, give a formal proof for the same. Otherwise, exhibit a counter-example and argue why it is in-fact a counter-example.

Solution

The given statement is true, i.e, if L is FO-definable then $\mathsf{en}(\mathsf{L})$ is MSO-definable.

L is FO-definable $\implies$ There is a DFA $\mathscr{A}$ accepting L (by FO to DFA construction).
$\implies$ There is a NFA $\mathscr{B}$ accepting en(L) (by homomorphic construction below).
$\implies$ There is a DFA $\mathscr{C}$ accepting en(L) (by subset construction of DFA from NFA).
$\implies$ There is an MSO-formula φ accepting en(L) (by DFA to MSO construction).
$\implies$ en(L) is MSO-definable.

Construction of NFA accepting en(L):
Given the DFA $\mathscr{A} = (Q, \Sigma, \delta_A, q_0, F)$ accepting L, we construct the NFA $\mathscr{B} = (Q', \Gamma, \Delta_B, q_0, F)$ accepting en(L) as follows:

For every transition in $\mathscr{A}$, $\delta_A(q_1, a) = q_2$ (where $q_1, q_2 \in Q$, $a \in \Sigma$, and $\mathsf{en}(a) = c_1 \dots c_k$),

- Case (i) - $k = 0$ [$\mathsf{en}(a) = \epsilon$], then add an ε-transition from q_1 to q_2. That is, $q_2 \in \Delta_B(q_1, \varepsilon)$

- Case (ii) - $k = 1$ [$\mathsf{en}(a) = c$], just replace a by c, i.e., $q_2 \in \Delta_B(q_1, c)$
- Case (iii) - $k \geq 2$, then create additional states $p_1, \ldots, p_{k-1}$ (no additional states if $k = 1$) and change the transitions to $p_1 \in \Delta(q_1, c_1)$, $\Delta(p_i, c_{i+1}) = \{p_{i+1}\}$ for all $i = 1, \ldots, (k-1)$ taking $p_k = q_2$.

Proof: For $w = c_1 \ldots c_n \in \Gamma^*$ (where $c_i \in \Gamma$),

$$\begin{aligned} w \in \mathsf{en}(L) &\iff \exists\, a = a_1 \ldots a_m \in L \text{ s.t. } w = \mathsf{en}(a) \\ &\iff \exists \text{ a path of states } q_0 \ldots q_m \text{ s.t. } q_m \in F \text{ and} \\ &\qquad \delta_A(q_i, a_{i+1}) = q_{i+1} \text{ for all } i = 0, \ldots, (m-1) \\ &\iff \exists \text{ a path of states } q_0 \ldots q_m \text{ s.t. } q_m \in F \text{ and} \\ &\qquad q_{i+1} \in \hat{\Delta}_B(q_i, \mathsf{en}(a_{i+1})) \text{ for all } i = 0, \ldots, (m-1) \\ &\iff \exists \text{ a path of states } q_0, q'_1, \ldots, q'_n \text{ s.t. } q'_n \in F \text{ and} \\ &\qquad q'_{i+1} \in \Delta_B(q'_i, c_{i+1}) \text{ for all } i = 0, \ldots, (n-1) \\ &\iff w \text{ is accepted by the NFA } \mathscr{B} \end{aligned}$$

The forward direction is quite straightforward, since we are adding all the required transitions in the construction of the NFA.

For proving the reverse direction, consider a path which lead to the acceptance of w by the NFA, and identify the old states (i.e., states not added during the construction) in the path (say, $q_0, \ldots, q_m$). Note that the initial state and final state are in Q. The property of the construction is that if there is a path from q_i to q_{i+1} containing only new states in between, then there exists a symbol $a \in \Sigma$ such that $q_{i+1} \in \hat{\Delta}(q_i, \mathsf{en}(a))$ following that path. (This is because the new states added are distinct for each transition, and hence each of the newly added states has a symbol in Σ associated with it.) With this observation, the reverse direction is also done.

Rubrics

- 0 marks: If L is FO definable, then $\mathsf{en}(L)$ be FO-definable. (As this statement is false)
- 1 mark: NFA by homomorphism is written but not explained (additional mark given for incomplete attempt to explain)
- 4 marks: NFA is formally defined but $L(\text{NFA}) = \mathsf{en}(L)$ is not proven
- 6 marks: DFA construction and proof are correct

No marks have been given for proofs on $\Sigma = \{a, b\}$ instead of all finite Σ.

Appendix: Automata Construction for Q2

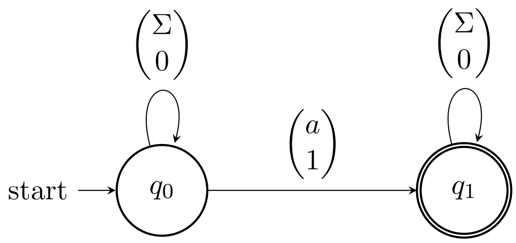

Figure 1: Automaton for $Q_a(x)$. 0.5 marks

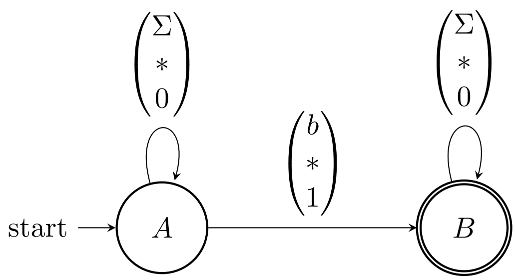

Figure 2: Automaton for $Q_b(y)$. We have added the second row for x, since we need to intersect this with $S(x, y)$ later. 0.5 marks

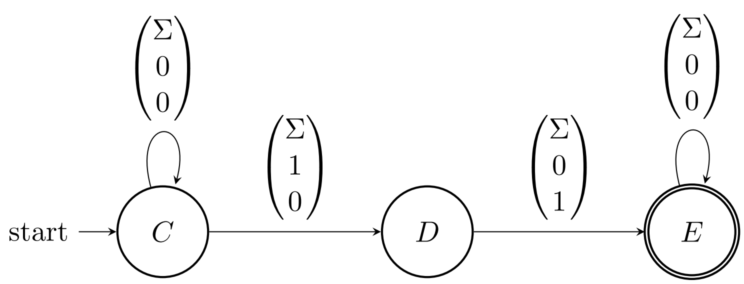

Figure 3: Automaton for $S(x, y)$. 0.5 marks

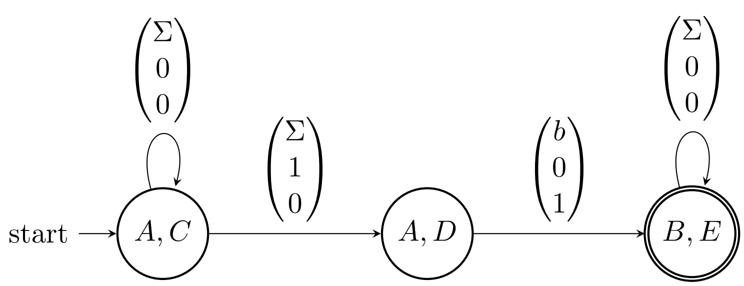

Figure 4: Automaton for $S(x, y) \wedge Q_b(y)$ after intersecting automata in Fig. 2 and Fig. 3. 1 mark

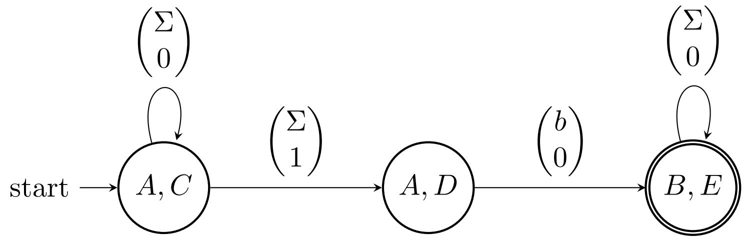

Figure 5: Automaton for $\exists y\, S(x, y) \wedge Q_b(y)$ after projecting the y-row in the automaton in Fig. 4. 0.5 marks

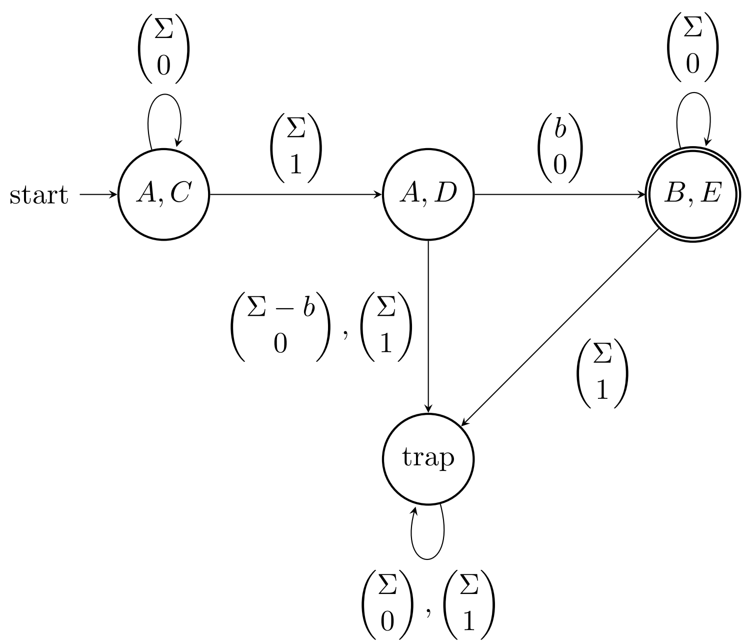

Figure 6: Complete DFA for $\exists y\, S(x, y) \wedge Q_b(y)$ after determinising Fig 5. 1 mark

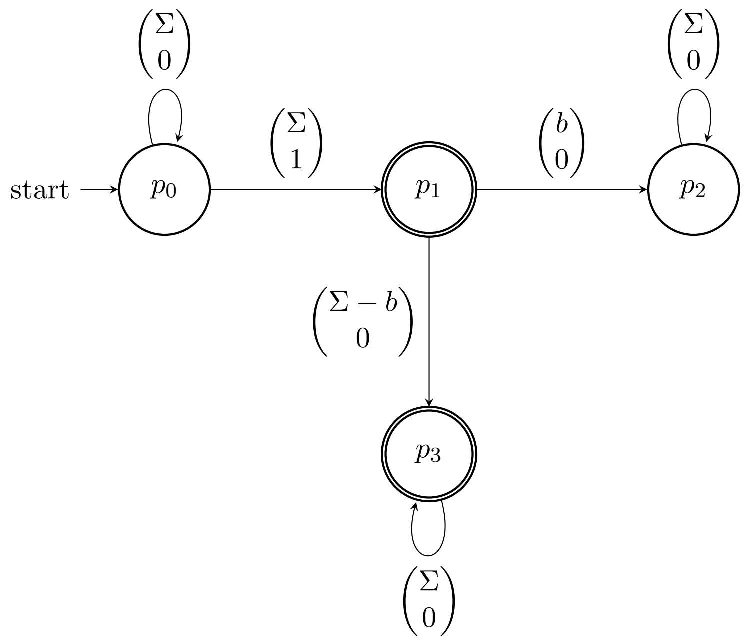

Figure 7: NFA for $\neg\exists y\, S(x,y) \wedge Q_b(y)$, after interchanging accepting and non-accepting states in Fig 6 and intersecting with $\binom{\Sigma}{0}^* \binom{\Sigma}{1} \binom{\Sigma}{0}^*$. 1 mark

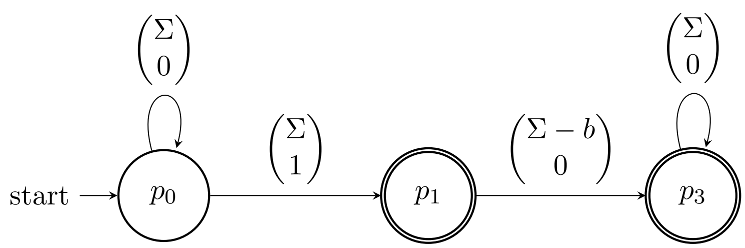

Figure 8: Minimal NFA for $\psi(x) = \neg\exists y[S(x,y) \wedge Q_b(y)]$ Not required

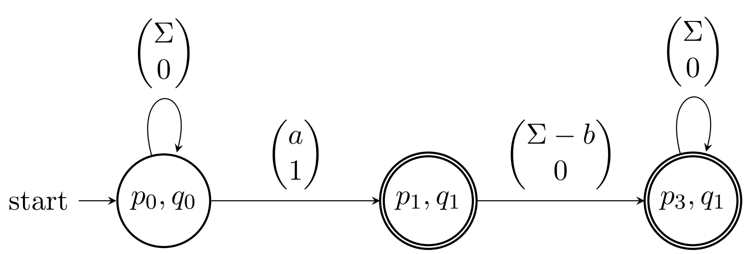

Figure 9: NFA for $Q_a(x) \wedge \psi(x)$ by intersection of Automaton 1 and Automaton 8. 1.5 marks

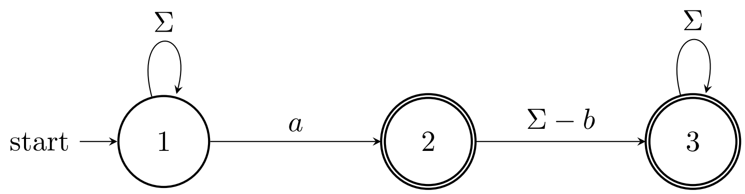

Figure 10: NFA for $\exists x\, Q_a(x) \wedge \psi(x)$ after projecting out the x-row in Automaton 9. 0.5 marks

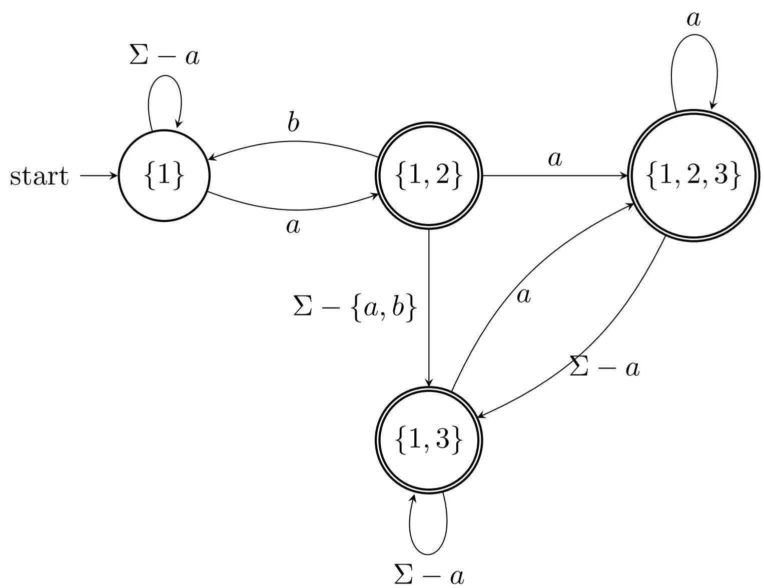

Figure 11: DFA for $\exists x\, Q_a(x) \wedge \psi(x)$ using subset construction. 2 marks

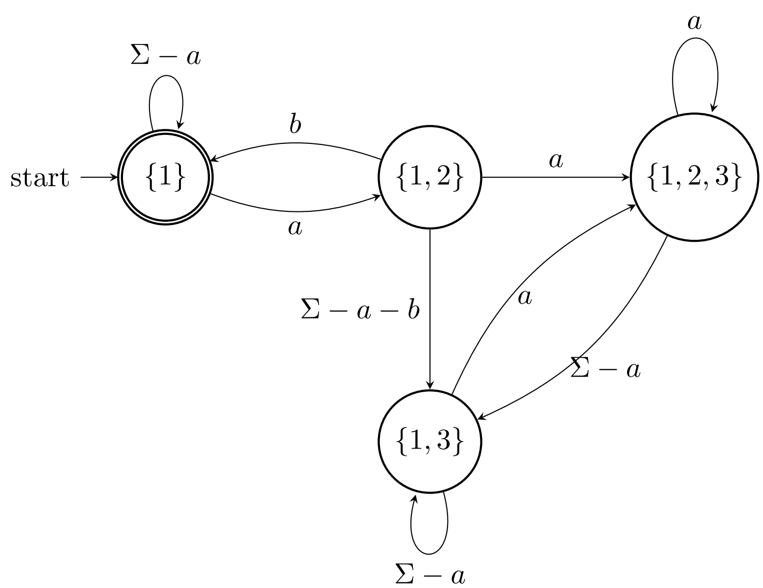

Figure 12: Final DFA for φ after negating the DFA in Fig. 11. 1 marks

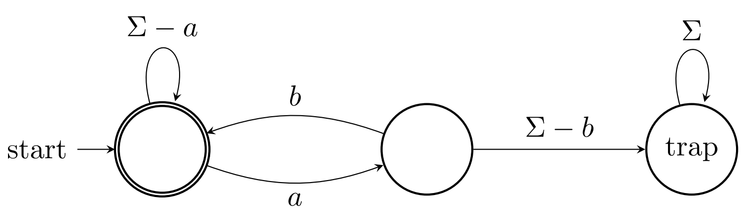

Figure 13: Minimal DFA for φ. Not required

Aliter: $\neg\exists x \neg(\neg Q_a(x) \vee \exists y(S(x,y) \wedge Q_b(y))))$

The marks indicated are the marks for just the automata. Marks are awarded for other steps as before.

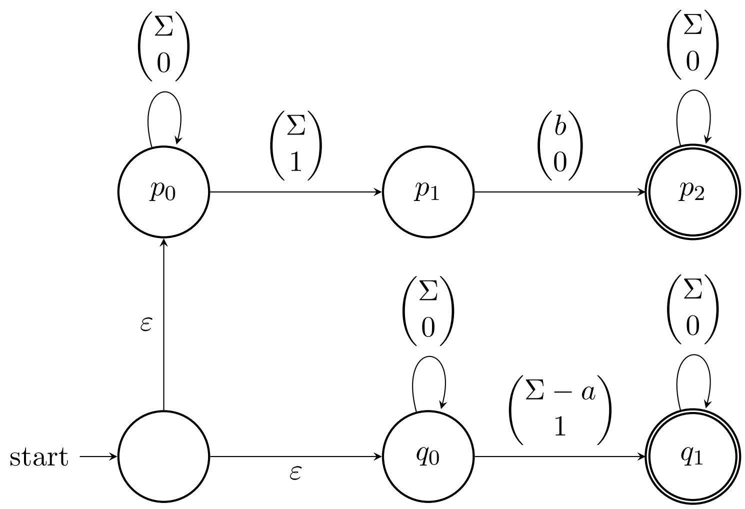

Figure 14: ε-NFA for $\neg Q_a(x) \vee \exists y\,(S(x,y) \wedge Q_b(y))$, using Fig 5. 0.5 mark

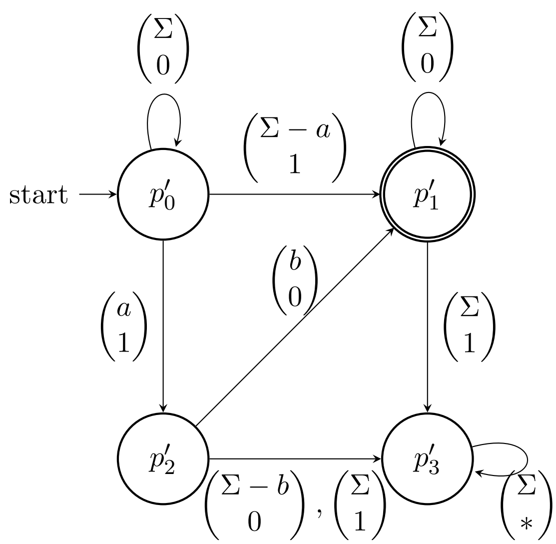

Figure 15: Minimal DFA for $\neg Q_a(x) \vee \exists y\,(S(x,y) \wedge Q_b(y))$ from Fig 14. 1.5 mark

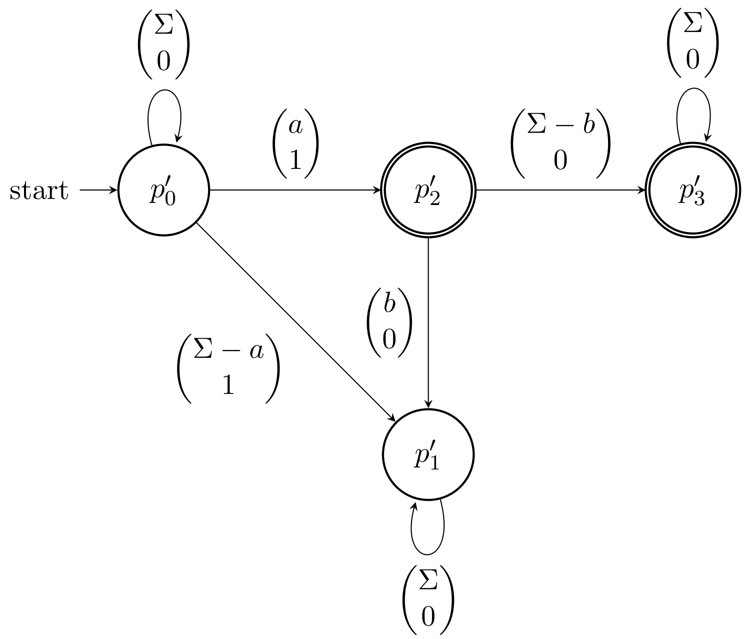

Figure 16: NFA for $\neg(\neg Q_a(x) \vee \exists y\,(S(x,y) \wedge Q_b(y)))$, after interchanging accepting and non-accepting states in Fig 15 and intersecting with $\binom{\Sigma}{0}^* \binom{\Sigma}{1} \binom{\Sigma}{0}^*$. After this, this will continue to Fig 10. 1.5 mark