Due: 17 March 2024

There are four problems listed below. To get full credit for this assignment you need to complete all of them.

If you are stuck or confused by any of the problems, please ask your tutor, lecturer or post to Piazza. You are allowed to discuss the problems with your peers and refer to online materials, but you are not allowed to share solutions or copy materials from any source. You may find the academic integrity rules at `https://academicintegrity.cs.auckland.ac.nz/`.

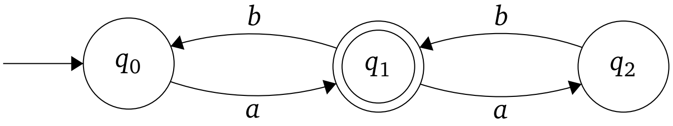
To get full marks you need to **show all working** unless a question explicitly states not to. You do not always need to provide fully formal proofs, but your arguments should be mathematically rigorous.

You should submit via Canvas a single PDF file containing the answers to the written questions. A handwritten submission is not acceptable. You may use a tool such as `https://madebyevan.com/fsm/` to draw automata diagrams.

Please try to submit your assignments no later than 5 min before the due time. Late assignments will not be accepted. If you need an extension due to illness or other personal circumstance, please email marc.vinyals@auckland.ac.nz as early as possible **before** the due date.

Best of luck, and enjoy the problems!

Problem 1 (10 marks)**.** Which language does the following NFA recognise? Explain why.



Problem 2 (10 marks)**.** Construct an NFA with at most 3 states that recognises the language

$$L = \{awb : w \in \{a,b,c\}^*\} \cup \{awc : w \in \{a,b,c\}^*\}$$

and explain why your construction works.

Problem 3 (10 marks)**.** Let $L = \{s\}$ be any language consisting of a single string. Determine which of the following statements are correct and prove or disprove them.

a) For every such L there exists a DFA with at most $|s| + 3$ states that recognises L.

b) For every such L, every DFA that recognises L has at least $|s| + 3$ states.

Problem 4 (10 marks)**.** Comment on the following conversation with ChatGPT. Which answers or parts of answers are correct? Which are errors? How can the errors be fixed? Give details.

> **User** Prove that the language $\{a^n b^m : n > m\}$ is regular.
>
> **ChatGPT** To prove that the language $L = \{a^n b^m : n > m\}$ is regular, we can construct a deterministic finite automaton (DFA) that recognizes it.
>
> Here's the informal idea of how we can construct such a DFA:
>
> a) Start in state q_0 to count the number of 'a's encountered.
>
> b) Once we encounter a 'b', transition to another state q_1 where we count the number of 'b's encountered.
>
> c) If we encounter more 'a's after 'b's, remain in q_1.
>
> d) If we encounter more 'b's after 'a's, move to a reject state.
>
> e) If we encounter 'a's and 'b's in equal numbers, move to a reject state.

Formally, we define the DFA as follows:

- $Q = \{q_0, q_1, q_2\}$, where q_0 is the start state, q_1 is the state where 'b's are being counted, and q_2 is the reject state.
- $\Sigma = \{a, b\}$.
- δ: Transition function defined as follows:
  - $\delta(q_0, a) = q_0$ (stay in q_0 when encountering 'a')
  - $\delta(q_0, b) = q_1$ (move to q_1 when encountering 'b')
  - $\delta(q_1, a) = q_2$ (move to q_2 if encounter 'a' after 'b')
  - $\delta(q_1, b) = q_1$ (stay in q_1 if encountering more 'b's)
- q_0 is the start state.
- $F = \{q_0, q_1\}$ (final states).

This DFA essentially counts the number of 'a's and 'b's, transitioning to a reject state if it encounters more 'a's than 'b's. Since it's a DFA, it always halts after reading the input string. If the number of 'a's is greater than the number of 'b's, it halts in the reject state; otherwise, it halts in an accepting state.

Thus, the language $L = \{a^n b^m : n > m\}$ is regular, as it can be recognized by a DFA.

User Prove that the language $\{a^n b^m : n > m\}$ is not regular.

ChatGPT To prove that the language $L = \{a^n b^m : n > m\}$ is not regular, we can use the Pumping Lemma for regular languages.

The Pumping Lemma states that if a language L is regular, then there exists a constant p (the pumping length) such that every string s in L with length at least p can be split into three parts, $s = xyz$, satisfying the following conditions:

a) $|xy| \leq p$

b) $|y| > 0$

c) For all non-negative integers i, the string xy^iz is also in L.

Let's assume that L is regular. Then, by the Pumping Lemma, there exists a pumping length p for L.

Consider the string $s = a^p b^{(p-1)}$. This string is in L because the number of 'a's (p) is greater than the number of 'b's ($p - 1$).

According to the Pumping Lemma, s can be split into three parts: $s = xyz$, where $|xy| \leq p$, $|y| > 0$, and for all non-negative integers i, the string xy^iz is also in L.

Since $|xy| \leq p$, the substring y consists only of 'a's. Let $y = a^k$ where $1 \leq k \leq p$.

Now, let's consider the string $s' = xy^2z$. The string s' will have more 'a's than 'b's, because we are doubling the number of 'a's but not the number of 'b's.

However, since $n > m$ in L, $s' = xy^2z$ should also belong to L. But this is not the case, because now the number of 'a's is greater than the number of 'b's, violating the condition of the language.

Therefore, we have reached a contradiction, implying that our assumption that L is regular is incorrect. Thus, the language $L = \{a^n b^m : n > m\}$ is not regular.