ASSIGNMENT 9 (DUE ON 10 OCTOBER 2025)

MATH2301, SEMESTER 2, 2025

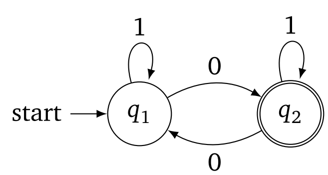
1. General remarks

(1) The assignment is due on gradescope.
(2) Please read the academic integrity policy for assignments. Remember that if you want an extension, you must ask at least 24 hours ahead of the deadline.
(3) The words "show" and "prove" are synonyms. You may not be used to writing formal mathematical proofs, which is OK. Write a justification in plain language that would convince the reader.
(4) If you are having trouble with any of the points, come and discuss with me in office hours. It is part of my job to help you understand this stuff, so please use my time!

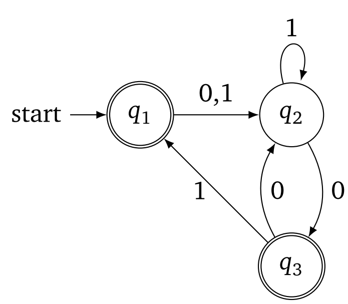
2. Problems

2.1. **Problem.** Convert the following DFAs into equivalent regular expressions. Show your work, but you need not give full justification.

(1)



(2)



2.1.1. *Solution.*

(1) First we add a dummy start and accept state.
   (a) Let us delete q_1. It has two incoming edges labelled ϵ and 0, and one outgoing edge labelled 0. It also has a self-loop labelled 1. So the new label on the edge $s \to q_2$ becomes 1*0. The new label on the loop $q_2 \to q_2$ becomes 1 | 01*0.
   (b) Next we delete q_2. The new label on $s \to a$ becomes

   $$1^*0(1 \mid 01^*0)^*.$$

   That is the final answer.
(2) First we add a dummy start and accept state.
   (a) Let us delete q_1. This changes the label on $s \to a$ to ϵ, the label on $s \to q_2$ to 0 | 1, the label on $q_3 \to q_2$ to 0 | 10 | 11, and the label on $q_3 \to a$ to ϵ | 1.
   (b) Next we delete q_3. This changes the label on $q_2 \to a$ to 0 | 01 and the label on the loop $q_2 \to q_2$ to 1|0(0 | 10 | 11).

(c) Finally we delete q_2. This changes the label on $s \to a$ to $\epsilon \mid (0|1)(1|0(0|10|11))^*(0|01)$. This is the final answer.

2.2. **Problem.** Show that the following languages are not regular using the pumping lemma and the Myhill-Nerode theorem (both). For the pumping lemma, assume (for contradiction) that there is a recognising DFA with n states. Clearly state the string you use for pumping. For the Myhill-Nerode theorem, clearly describe an infinite set of non-equivalent strings.

(1) $L = \{0^k 1^l \mid k < l\}$.
(2) $L = \{0^n \mid n \text{ is a square.}\}$

2.2.1. *Solution.*

(1) We first give a solution using the pumping lemma.
Assume that there is a recognising DFA with n states. Consider the string $0^{n+1}1^{n+1}$, which the DFA must reject. As the DFA reads the substring 1^{n+1}, one of its states must repeat, resulting in a directed cycle. If we repeat this cycle, the outcome given by the DFA does not change. But the string becomes $0^{n+1}1^{m+1}$ for some $m > n$, which the DFA must accept. This is a contradiction.
We now give a solution using the Myhill-Nerode theorem. Consider the strings $0, 00, 000, 0000, \cdots$. If $m < n$, then we see that $x = 0^m$ and $y = 0^n$ are distinguished by $z = 1^n$, because $xz \in L$ but $yz \notin L$. So the strings of the form 0^m are pairwise inequivalent.
(2) We first give a solution using the pumping lemma.
Assume that there is a recognising DFA with n states. Consider 0^{n^2}, which must be accepted. While the DFA reads it, it must repeat a state and create a directed cycle. The smallest possible directed cycle has length at most n. If we repeat it, we get a string of the form 0^{n^2+m} where $0 < m \leqslant n$. Note that the next perfect square is 0^{n^2+2n+1}. So the DFA must reject 0^{n^2+m}; but repeating a cycle does not change the outcome. This is a contradiction.
We now give a solution using the Myhill-Nerode theorem. Consider the strings 0^{n^2} for $n = 1, 2, \cdots$. If $m < n$, then $z = 2m + 1$ distinguishes 0^{m^2} and 0^{n^2} (why?) So we have infinitely many pairwise inequivalent strings.

2.3. **Problem.** Let L be the language described by $(0|1)^*11$.

(1) Find the Myhill-Nerode equivalence classes for L (give one string in each equivalence class).
(2) Construct a DFA that recognises L with exactly as many states as the number of equivalence classes.

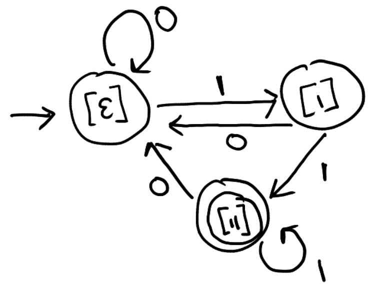
2.3.1. *Solution.* Membership in L only depends on the last two letters of the string. Using this, we see that if z has length 2 or more, then xz and yz are either both in L or both not in L. So the only z that helps distinguish strings are $z = \epsilon, 0, 1$. Using this, we see that:

(1) If x ends in 11, then $x \sim 11$.
(2) If x ends in 01, then $x \sim 1$.
(3) If x ends in 0, then $x \sim \epsilon$.

So, every string is equivalent to one of: $\epsilon, 1, 11$. To see that these represent are distinct equivalence classes, we note that

(1) $z = \epsilon$ distinguishes the first two from 11.
(2) $z = 1$ distinguishes ϵ from $1, 11$.

To construct the optimal automaton, we take three states labeled by the equivalence classes $[\epsilon], [1], [11]$ and draw the transitions $0\colon [s] \to [s0]$ and $1\colon [s] \to [s1]$. The start state is $[\epsilon]$ and the accept state is $[11]$.



[ε]
[1]
[11]