ASSIGNMENT 7 (DUE 26 SEPTEMBER 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.** Let $\Sigma = \{0, 1\}$. For each language L described below, write down a regular expression r such that $L(r) = L$. That is, the strings that match r are exactly the strings in L. Be careful to make sure that nothing else matches the regular expression you write down! Justification is not required.

(1) $L = \{w \in \Sigma^* \mid w \text{ starts with a } 1\}$
(2) $L = \{w \in \Sigma^* \mid w \text{ all the ones in } w \text{ are next to each other in a single block}\}$
(3) $L = \{w \in \Sigma^* \mid w \text{ contains an even number of zeroes}\}$

2.1.1. *Solution.*

(1) $r = 1(1 \mid 0)^*$.
(2) $r = 0^*1^*0^*$.
(3) $r = 1^* \mid (1^*01^*01^*)^*$.

2.2. **Problem.** Let $\Sigma = \{a, b, c\}$. For each regular expression r written below, describe in words the language $L(r)$. Justification not required.

(1) $r = (\epsilon|bc|c)(abc)^*(\epsilon|a|ab)$.
(2) $r = ((b|c|\epsilon)^*a(b|c|\epsilon)^*a(b|c|\epsilon)^*a(b|c|\epsilon)^*)^*$

2.2.1. *Solution.*

(1) This is the language such that the symbols a, b, c appear in cyclic order in any word in the language.
(2) This is the language L such that $w \in L$ is either the empty word, or contains $3k$ instances of the symbol a, where $k \geqslant 1$.

2.3. **Problem.** Let $\Sigma = \{0, 1\}$ and L be the language

$$L = \{w \mid \text{ the number of occurencess of 01 in } w \text{ is equal to the number of occurencess of 10.}\}$$

For example, the word 010 is in L because it has one occurences of 01 and one of 10. The word 01101 is not in L because it has 2 occurences of 01 but only one of 10. Does there exist a regular expression r with $L(r) = L$? If yes, find one. If not, explain why not.

2.3.1. *Solution.* The description of the language can be reinterpreted as follows. If a word either has only 0s or only 1s, then it is in L. Otherwise, if a word has both 0s and 1s, then it can be in L if and only if it begins and ends with the same letter. The reason is that, for example, if the word begins with a 0, then it consists of some number of 0s followed by some number of 1s, followed by 0s, and so on. Every switch from 0s to 1s introduces the substring 01, and every switch from 1s to 0s introduces the string 10, and this alternates. If the string begins with 0 and ends with 1, there will be one more occurrence of 01 than of 10. Similarly if the string begins with 1 and ends with 0, there will be one more occurrence of 10 than of 01.

So here is a regular expression: $0^*|1^*|0(0|1)^*0|1(0|1)^*1$.

2.4. **Problem.** Let $L \subseteq \Sigma^*$ be a language. The *complement of* L, denoted L^c, is the complement of L in Σ^*. That is, for every $w \in \Sigma^*$, we have $w \in L^c$ if and only if $w \notin L$.

(1) Given a DFA M recognising a language $L = L(M)$, explain in words how to construct a DFA M' such that $L(M') = L^c$.

(2) Construct a DFA recognising the following language:

$$L = \{w \in \Sigma^* \mid \text{every odd position of } w \text{ is } 1\}.$$

(Assume that we start indexing at 1, not 0.) Justification not required.

(3) Now use your method from the first part to draw a DFA for the complement of the language L above. Justification not required.

   (a) Given a DFA M recognising a language $L = L(M)$, explain in words how to construct a DFA M' such that $L(M') = L^c$.

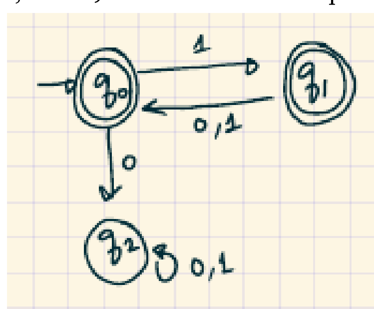
2.4.1. *Solution.*

(1) Change every accept state to a start state, and vice-versa.

(2)

$$L = \{w \in \Sigma^* \mid \text{every odd position of } w \text{ is } 1\}.$$

(Assume that we start indexing at 1, not 0.) Justification not required. Here is an answer.



(1)