COMP 330 Winter 2021
Assignment 2
Due Date: 4^{th} February 2021

Prakash Panangaden

21*st* January 2021

There are **5** questions for credit. There are four questions at the end. The first of them requires some algebra. The second is a bit harder than the usual homework but not impossible; it is worth looking at. Don't even look at the third one, it is impossible. The fourth is a puzzle. If you can't do it, it does not mean anything about how well you are understanding the material. If you like mathematical puzzles you might enjoy this one. The homework is due on myCourses at 5pm.

Question 1[20 points]
Give regular expressions for the following languages over $\{a, b\}$:

1. $\{w|w$ contains an even number of occurrences of $a\}$

2. $\{w|w$ contains an odd number of occurrences of $b\}$
3. $\{w|$ does not contain the substring $ab\}$

4. $\{w|$ does not contain the substring $aba\}$

Try to make your answers as simple as possible. We will deduct marks if your solution is *excessively* complicted.

Question 2[20 points]
Suppose that you have a DFA $M = (S, \Sigma, s_0, \delta, F)$. Consider two distinct states s_1, s_2 *i.e.* $s_1 \neq s_2$. Suppose further that *for all* $a \in \Sigma$ $\delta(s_1, a) = \delta(s_2, a)$. Show that for any *nonempty* word w over Σ we have $\delta^*(s_1, w) = \delta^*(s_2, w)$.

Question 3[20 points]
Show that the following languages are not regular by using the pumping lemma.

1. $\{a^n b^m a^{n+m} | n, m \geq 0\}$,
2. $\{x | x = x^R, x \in \Sigma^*\}$, where x^R means x reversed; these strings are called *palindromes*. An example is *abba*, a non-example is *baba*.

Question 4[20 points] Show that the following languages are not regular by using the pumping lemma.

1. $\{x \in \{a, b, c\}^* || x| \text{ is a square.}\}$ Here $|x|$ means the length of x.
2. $\{a^{2n}b^n\}$.

Question 5[20 points] We are using the alphabet $\{0, 1\}$. We have a DFA with 5 states, $S = \{s_0, s_1, s_2, s_3, s_4\}$. The start state is s_0 and the only accepting state is also s_0. The transitions are given by the formula

$$\delta(s_i, a) = s_j \text{ where } j = i^2 + a \mod 5.$$

Draw the table showing which pairs of states are inequivalent and then construct the minimal automaton. Remember to remove useless states right from the start, before you draw the table. I am happy with a drawing of the automaton.

Extra Question 1[0 points]
Let M be any finite monoid and let $h : \Sigma^* \to M$ be a monoid homomorphism. Let $F \subseteq M$ be any *subset* (not necessarily a submonoid) of M. Show that the set $h^{-1}(F)$ is a regular language. This means you have to describe an NFA (or DFA) from the given M, F and h[8 points]. Show that *every* regular language can be described this way.[12 points]

Extra Question 2[0 points] We define the Hamming distance between two strings w, x *of the same length* to be the number of places where they differ. If the strings have different lengths we say that their Hamming distance is infinite. We write is as $H(x, y)$. For example, $H(000, 010) = 1$ and $H(0000, 1001) = 2$. Given a set of words A and a positive integer k we define the new set $N_k(A)$ as follows

$$N_k(A) = \{x | \exists y \in A \text{ such that } H(x, y) \leq k\}.$$

For example $N_1(\{000, 111\}) = \{000, 001, 010, 100, 111, 110, 101, 011\}$ and $N_2(\{000\}) = \{000, 001, 010, 100, 110, 101,$
Of course, these are only examples and my definition of $N_k(A)$ is perfectly valid when A is an infinite set. **Prove** that if L is regular then $N_2(L)$ is regular. What happens to $N_k(L)$?

Spiritual growth[0 points] In extra question 1, we showed how one could have defined regular languages in terms of monoids and homomorphisms instead of in terms of DFA. Given a regular language L, we can define an equivalence relation on words in Σ^* as follows:

$$x \equiv_L y = \forall u, v \in \Sigma^*, uxv \in L \iff uyv \in L.$$

It is easy to see that this is a congrence relation (with respect to concatenation). If we quotient by this equivalence relation we get a monoid called the syntactic monoid of the language L. The syntactic monoid is finite iff L is regular. Now what can you say about the language if the monoid happens to be a group? What if it is not only not a group but contains no subgroup? Yes, a monoid that is not a group could have a submonoid which is a group.

Pure puzzle[0 points] Design an **NFA** K with n states, over a one-letter alphabet, such that K rejects some strings, but the *shortest* string that it rejects has length *strictly* greater than n.