COMP 330 Fall 2021
Assignment 2
Due Date: 19^{th} October 2021

Prakash Panangaden

$5th$ October 2021

There are **5** questions for credit. The homework is due on myCourses at 5pm.

Question 1[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.

Question 2[20 points]
Are the following statements true or false? Justify your answer in each case. We have some fixed alphabet Σ with at least two letters. In the following A and B stand for languages, *i.e.* subsets of Σ^*.

- If A is regular and $A \subseteq B$ then B must be regular. [3]
- If A and AB are both regular then B must be regular. [7]
- If $\{A_i | i \in \mathbb{N}\}$ is an infinite family of regular sets then $\bigcup_{i=1}^{\infty} A_i$ is regular. [5]
- If A is not regular it cannot have a regular subset. [5]

Question 3[20 points]
Consider the language $L = \{a^n b^m | n \neq m\}$; as we have seen this is not regular. Recall the definition of the equivalence $\equiv_L$ which we used in the proof of the Myhill-Nerode theorem. Since this language is not regular $\equiv_L$ cannot have finitely many equivalence classes. Exhibit explicitly, infinitely many distinct equivalence classes of $\equiv_L$.

Question 4[20 points] Describe an algorithm that given two different regular expressions R_1 and R_2 decides whether $R_1 \subseteq R_2$. The description should be high-level and at the level of detail shown in the example I posted on the website. **I will deduct marks for excessive low-level details and I will give you zero if you submit code.**

Question 5[20 points] Let D be the language of words w such that w has an even number of a's and an odd number of b's and does not contain the substring ab.

1. Give a DFA with *only five* states, including any dead states, that recognizes D.
2. Give a regular expression for this language.

Extra Question [0 points]
This question is for fans of algebra. Not doing it will not affect your understanding of the material. 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. Show that *every* regular language can be described this way.

Spiritual growth[0 points] In the extra question above, 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 (prove it!). 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. This is a deep and difficult question; the result is called Schutzenberger's[1] theorem.

[1] Pronounced the French way.