COMP 330 Autumn 2015
Assignment 2
Due Date: 8^{th} October 2015

Prakash Panangaden

24^{th} September 2015

There are **5** questions for credit, one for your spiritual growth [Q6] and one [Q7] for pure puzzle-solving fun. There is nothing to be worried about if you cannot do Q7 but people with a taste for elementary number theory may find it fun. The homework is due in class at the beginning of the class.

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

1. $\{w|w$ begins and ends with an $a\}$
2. $\{w|$ every odd position is a $b\}$ [I did **not** say that every even position is a b.]
3. $\{w|w$ contains at least two as and at most one $b\}$
4. All strings except the empty string.

Question 2[20 points] For each of the following regular expressions, describe an NFA that accepts the language described by the regular expression:

1. $\epsilon + a(a+b)^* + (a+b)^*aa(a+b)^*$
2. $[ba + (a+bb)a^*b]^*$

Alternate Question 2[20 points] Suppose that S, T are regular expressions and X is a variable that ranges over regular expressions. We write equations involving regular expressions using X. Consider the following

$$X = SX + T.$$

Suppose that $\varepsilon \notin S$, show that $X = S^*T$ is the *unique* solution of the equation. The easy part is showing that it is a solution (2 points). The hard part (18 points) is showing that it is the unique solution. Why did we insist that $\varepsilon \notin S$?

Question 3[20 points] Show that following languages are regular? The alphabet is $\{a, b\}$.

1. $\{xwx^R | x, w \in \Sigma^*, |x|, |w| > 0\}$ and x^R means x is reversed. For this one it suffices to give me a regular expression. You do not have to prove that it is correct.
2. $\{a^n b^m | n, m > 0, n - m = 0 \bmod 3$. For this one I would like to see an automaton.
3. $\{a^n b^m | n, m > 0, n + m = 0 \bmod 3\}$. For this one you must describe an automaton.

You do not have to prove that the automata are correct and I am happy with a picture of the automaton.

Alternate Question 3[20 points] Suppose that L is *any* language over the alphabet Σ. We define the derivative of L with respect to a symbol $a \in \Sigma$ by

$$\partial_a L = \{w \in \Sigma^* \mid a \cdot w \in L\}.$$

Show that the derivative of a regular language is regular. [5 points] Thus, we can define the derivative of a regular *expression* since we know that it gives a regular expression. Thus we can define a *formal calculus* of regular expressions. We need one more item to do so. We define the function ν which maps regular expressions to regular expressions according to the following formula

$$\nu(R) = \begin{cases} \varepsilon \text{ if } \varepsilon \in R \\ \emptyset \text{ otherwise.} \end{cases}$$

It is straightforward to define ν by induction on regular expressions. I do not want you to write that up, I assume you can do it easily. What I want is an inductive definition of ∂_a on regular expressions. The interesting cases are $*$ and concatenation for which you will need to come up with an analogue of Leibnitz' rule. I just want you to write up the rules, you do not have to prove anything. [15 points]

Question 4[20 points] Consider the DFA with the following structure. There are 7 states labelled $\{A, B, C, D, E, F, G\}$ with A as the start state and $\{B, D\}$ as the accept states. The input alphabet is $\{0, 1\}$. The transition table is:

A	0	A	E	0	E
A	1	B	E	1	D
B	0	B	F	0	G
B	1	C	F	1	F
C	0	C	G	0	F
C	1	D	G	1	F
D	0	D			
D	1	E			

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.

Alternate Question 4[20 points] Suppose that x, y are both non-empty words such that $xy = yx$. Show that there is a non-empty word z and positive integers n, m such that $x = z^n$ and $y = z^m$.

Question 5[20 points] Consider the LTS with the following states:

$$S = \{s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9\}$$

and actions $\{a, b, c\}$. The transitions are

s_0	a	s_1, s_2, s_3
s_1	b	s_4
s_1	c	s_5
s_2	b	s_6
s_3	c	s_7
s_3	b	s_8, s_9

Which states are bisimilar? Give the bisimulation-minimal description of the LTS.

Alternate Question 5[20 points] Given an alphabet Σ with two or more letters show that

$$\forall x, y \in \Sigma^*, xy = yx \text{ if and only if } \exists z \in \Sigma^*, \text{ such that } z^2 = x^2y^2.$$

Question 6[0 points] What would happen to automata theory if we allowed infinite words? Of course there is no question of "recognizing" an infinite string in a finite amount of time but can one still come up with a mathematically meaningful notion of "regular?" What could it mean to accept a string? Clearly we cannot ask the machine to end up in an accept state since the word never ends. Is there a nice topological characterization of the regular languages in this setting? Believe it or not, this theory actually exists and has numerous applications in industry.

Question 7[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.