Problem 1 [3 points]
Prove that given any NFA($N$) with a transition function $\delta_N$, such that $|\delta_N(q,a)| = 1$, for all states $q$ and for all input symbol $a$, if a DFA($D$) is constructed from $N$ by using subset construction then $D$ has all the transitions of $N$ plus some additional transitions to a new dead state (say $\Phi$) wherever $N$ is missing a transition for a given state and input symbol (refer to the box on p.67 of your text book).

Solution:
Here you are supposed to argue that the subset construction of such and NFA gives the said DFA. This goes something as following.
Consider the process of subset construction. For each transition from a state we construct a set of all reachable states. The claim is that these sets will all be of unit size. Notice that there are no $\varepsilon$-transitions so ECLOSE will also give back the same set. For all those inputs for which a transition is missing from the NFA we reach the empty set. This is actually the Dead or Crash state. Thus we get the desired DFA.

Problem 2 [5 points]
In order to prove a language to be regular you are always required to give an automata, a regular expression, or use the closure properties of regular languages to show its equivalence with obtain an already known regular language. Why can’t we use pumping lemma for regularity to prove so?

Solution:
The simplest argument here is that the statement of pumping lemma is an implication, i.e. if P then Q type. The Hypothesis is that “if a language is regular” and the conclusion is that “then it can be pumped OR there exist a number n, s.t. all words $w$ of the language can be decomposed into some $x,y,z$ s.t …..”.
The statement “If P then Q” does not mean “If Q then P”. So if a language can be pumped then it does not necessarily mean that it’s a regular language. That could easily imply that the language is just a subset of a regular language.

Problem 3 [5+7+11= 25 points]
Following languages may or may not be regular. In case you think they are regular, just give a basic idea how you are going to construct an FA or a regular expression for them. Otherwise use pumping lemma for regular languages to show that these are non-regular languages.

1. $L = \{xyz \mid x,y,z \in \{0,1\}^* \land |x| = |z| \text{ and number of zeros in } x$ and $z \text{ are equal}\}$
The way this language is defined is a slightly confusing; but you can obtain any string from the alphabet using this definition of the language. This is obvious if you keep x and z to be empty, which obviously satisfies the constraints on them. There are no restrictions on y, so this just becomes the language of $\Sigma^*$. Thus the language is regular.

2. $L = \{x^2x \mid x \in \{0,1\}^*\}$
Just pick $w = 0^n20^n$; this would force y to be consist of zeros coming before 2. Then pump this y zero or greater then 1 number of times and your number of zeros to the left and right of the central 2 will not be the same; thus giving you a string that is not acceptable by the language.

3. $L = \{a^n \mid n \text{ is not a prime}\}$
If you try (for any arbitrary decomposition of this string into x, y and z) to pump y so that the whole length of the string comes out to be a prime, then this strategy would be extremely difficult (or maybe impossible) to work. But proving that the complement of this language is non-regular is very easy (given in the book). However in order to get full credit you are required to reproduce that proof here. Then you can use the fact that the complement of a non-regular language is also non-regular (problem 7 part a) to say that this language is non-regular.

**Problem 4** [7 points]
Suppose you are given an automaton $A$ which generates the language $L$, that is $L(A) = L$. Now let us perform the following transformations on $A$ to obtain a new automaton $A'$.

1. Change all the accepting states into non-accepting states.
2. Change all the non-accepting states into accepting states.

Prove that, regardless of the fact that $A$ is deterministic or non-deterministic, $L(A') = L^C$.

(The argument required here would be similar to one used in Homework 01, problem 02).

**Proof (DFA case)—NFA case can be easily obtained by giving an equivalent DFA and using the same argument.**

$L(A') \subseteq L^C$
Consider any string $x$ in $L(A')$. This means that when $A'$ processes $w$, it ends up in an accepting state of $A'$. This means that when $A$ processes $w$, it ends up in a rejecting state of $A$. Then, by our construction this means that $w$ is not in $L(A)$, which implies $w$ is not in $L$. So $w$ must be in $L^C$, and we are done.

$L^C \subseteq L(A')$
Consider any string $w$ in $L^C$. This means that $w$ is not in $L$, which means $w$ is not in $L(A)$. When $A$ processes $w$ it ends up in a rejecting state of $A$. But this leads to the fact that when $A'$ processes $w$, it ends up in an accepting state of $A'$.
This means that $w$ is in $L(A')$, and we are done.

**Problem 5** [5+13+11 = 29 points]
Give a regular expression for each of the given languages. You may make use of any of the closure properties of regular languages. To get full credit argue informally (and...
briefly) why the transformations would give you the correct solution that. Show all workings (you may want to go through sections 4.2.1 and 4.2.2 of your textbook).

1. Give $L^c$ where $L = \{w \mid w$ does not contain the substring 011$\}$
2. Give $L^R$ and $L^c$ when $L = \{w \mid w$ does not begin and end with the same letter, and contains odd number of 1s$\}$
3. Give $L^R$ when $L^c = \{w \mid w$ contains even 0s and odd 1s$\}$

Solution to this problem will be uploaded separately.

Problem 6 [7+10 points]
You can use the closure properties of regular languages to prove any language $L$ to be non-regular, given a non-regular language $L'$. Start by assuming that $L$ is regular, and then transform $L$ to $L'$ by applying closure properties. This leads to the contradiction that $L'$ is also regular. Use this technique to prove the following to be non-regular languages, where $L$ is given as

$L = \{0^n1^n \mid n \geq 0\}$

(Do not use pumping lemma here)

Solution:
The strategy for solving these questions is simple. Assume the language to be regular and then apply some closure property which would lead to the given non-regular language i.e. $0^n1^n$. A common mistake is to start with this given non-regular language and trying to use the properties to transform this into the given language. But for this you would require to prove that the properties you are using are closed under non-regular languages as well (which is not true for all the properties)

1. $L = \{0^n1^n2^m \mid n \geq m \geq 0\}$

Use the Homomorphism property with the functions; $H(0) = 0; H(1) = 0; H(2) = 1$; This will transform this into $0^n1^n$ which is the given non-regular language, so the initial assumption was wrong.

OR

Just take the intersection of this language with the regular expression $0^*1^*$. The only strings that are common are those in which there are zero occurrences of the alphabet 2. This can only happen when $n$ equals $m$. Thus this intersection will give $0^n1^n$; which is the given non-regular language.

2. $L = \{0^n1^n \mid$ where $n$ and $m$ are co-prime$\}$

Simple homomorphism will work in this. We first create two new regular languages using the following functions

$H_1(0) = 0; H_1(1) = 0$;
This gives us $L_1$ which is regular (due to our original assumption).
Similarly we can get $L_2$ by using
$H_2(0) = 1; H_2(1) = 1$;

Now concatenating these two languages we get a new regular language which is actually $0^n1^n$ hence a contradiction.

Problem 7 [7+10=17 points]
Prove or disprove the following statements:

a) “If L is a non-regular language then its complement $L^C$ also non-regular”.

The complement of any non-regular language is always non regular. The proof is by contradiction. Suppose that given a non regular language L such that its complement is regular given by some regular expression R. Consider the language given by $L^C( R)$. This surely must be regular due to closure of regular languages under complementation. But we know complementation is idempotent. Thus we have a contradiction.

b) “If $L_1$ and $L_2$ are two non-regular languages then $L_1 \cup L_2$ is also non-regular”. (Hint: solve part a first)

This statement is incorrect and a simple counter-example can be constructed easily. Given a non-regular language L obtain its complement $L^C$, which is also non-regular (by the fact proved earlier). Now take the union of L and $L^C$. This is actually $\Sigma^*$ (the union of any set and its complement is the universal set) which is definitely a regular language.