1. (30 pts). Problem 1.46 a) , c) on page 90 and Problem 1.71 b) on page 93of the textbook.

1.46 a)

we want to prove: Some language Q defined as $\{a^n 1^m 0^n | m, n \geq 0\}$is not a regular language

We will utilize a proof by contradiction in assuming that Q is a regular language

Assume Q is a regular language. If Q is a regular language then there is a finite automaton which recognizes any word of Q.

We know for a word to be in Q it must be constructed by $\{a^n 1^m 0^n | m, n \geq 0\}$ if we let s be some word 0001000 then we can apply the pumping lemma and let p be the length of the pumping lemma

we will use p to define s as $x^{p-1}y^k z^p$ where k is greater than 1 where x is equal to 00 , y is equal to 0, and z is equal to 1000. Since s is $\in$ Q the pumping lemma guarantees it can be broken down into xy^iz where i is greater than or equal to 0. But if k is equal to say 10 in a similarly constructed string, it is not in Q. Therefore by the pumping lemma, Q is not a regular language since we found a string of length p which could not be broken down to satisfy the conditions required by the pumping lemma to be in the class of regular languages. Therefore we have proved that the language in problem 1.46 is not regular.

——————————————-

1.46 c)

To prove that L = $\{w | w$ is not a palindrome$\}$

is not a regular language.

We will use the closure of the class of regular languages under complement property. We will utilize a proof by contradiction in assuming that some language Q is a regular language

Since Q is a regular language its compliment and itself are in the same class of regular languages.

let Q be the compliment of L. That is Q = $\{w | w$ is a palindrome$\}$. since we know if a word is either a palindrome or not a palindrome; that is the set of palindrome and not a palindrome are exhaustive and exclusive of all words.

Now we begin our proof by contradiction

Assume L is a regular language. If L is a regular language then there is a finite automaton which recognizes any word of L. We can also construct an automaton which accepts any word in Q as per the definition of a regular language to be closed under the compliment operator.

Therefore there is a word which is in the language Q which is the Complement of a word in language L.

We know for a word to be in Q it must be a palindrome. we can define some palindrome s as some word $\{w = abc\}$ — where a consists of 4 0's and c consist of a 1 followed by 4 0's and b is a single 1. We can now apply the pumping lemma and let p be the length of the pumping lemma we will use p to define s as $x^{p-1}y^kz^p$ where k is greater than 1 where x is equal to a, y is equal to b, and z is equal to c. Since s is $\in$ Q the pumping lemma guarantees it can be broken down into xyz even if k = 0. But if k is equal to 0 in a similarly constructed string, it is not a palindrome. Therefore by the pumping lemma, Q is not a regular language since we found a string of length p which could not be broken down to satisfy the conditions required by the pumping lemma to be in the class of regular languages. Since we know regular languages are closed under the compliment operator, we know that if Q is not a regular language, and we know L is Q's compliment, then we know L is not a regular language. Therefore we have proved that the language in problem 1.46 is not regular.

1.71

a)

since u can be any element of the alphabet any number of times. we can simply apply the pumping lemma where y equals u. then there is always a finite way to count the number of substrings y whatever element y is. Similarly we can count every zero before we get to y and compare it to the count of zeros after y.

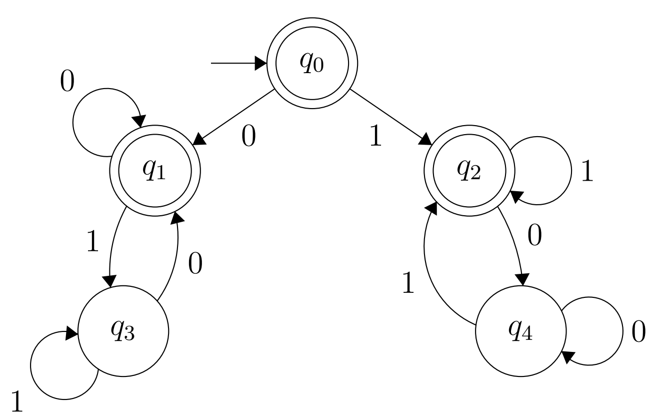
---

b)

if we try to apply the pumping lemma to the language B we will be unable to construct a finite automaton which will accept B. The reason is that we do not know if u will be a 1 or a 0. so then counting the number of 0's is futile since we have no way to tell if the one came directly in the middle of them or slightly to the left of center. Without memory once we entered the first state of being transitioned via a 0 we would be unaware how to match the number of zeros on the right side to the number of zeros on the left since we do not know if u was a 1 or a 0.

---

2. D is a regular language if we can construct a finite automaton which recognizes it.



---

3. problem 1.54

   a)

   To prove that F

   is not a regular language.

   We will use the closure of the class of regular languages under union property. We will utilize a proof by contradiction in assuming that some language Q is a regular language

   Since Q is a regular language its composed of the union of language F with itself so then both are in the same class of regular languages.

   Now we begin our proof by contradiction

   Assume L is a regular language. If L is a regular language then there is a finite automaton which recognizes any word of L. We can also construct an automaton which accepts any word in Q as per the definition of a regular language to be closed under the union operator.

   Therefore there is a word which is in the language Q which is the Union of a two words in language L.

   We know for a word to be in F it must be some word s as some word $\{w = abbbbb...bcccc...c\}$ — We can now apply the pumping lemma and let p be the length of the pumping lemma as long as y can be selected to contain 2 or more B's, then the pumping lemma can not show that a word can be pumped.

   So then, y^i is only true when i equals 1. since once i equals more than 1 there will be more b's then c's and thus F will not accept it.

   if a word cannot be pumped such that F will accept it no matter what i is. Then F is not a regular language by contradiction

   b)

   The pumping lemma will work on any string s in language F so long as we always select y to contain only a's OR if there are either 0 a's or 2 or more a's in the string s. then we can easily select y to be either any number of b's before c's or any number of c's after b's

   c)

   The reason the above two statements do not contradict each other is because the pumping lemma only shows that all regular languages possess the property such that all of the conditions of the pumping lemma hold for it. That is: if a prioperty does not hold for the language then the language is not regular. It does not imply anything to the converse however– meaning: if the property does hold we still know nothing about whether or not it is regular.

   ---

4. 1.61

   to construct a DFA we must have a transition for each input of the alphabet from every state.

Since every time at the point in time when a machine receives the input a, it must begin counting to see if it receives k more elements before the end of the string to know if the word is accepted. Similarly if it recieves an a on a subsequent input it must begin counting from that state. The machine does this by having a sequence of states of size k from each state where it just received an a. This sequence must include all possibilities of a or b. and then from each of the a's there must be a similar list, because the machine (when it receives an a) has no idea how many more elements it will receive before the input word ends. but it must have the capability to count up to exactly k more input elements before it can go to a state where it will not be able to reach an accept state. '

The result of this implication is a machine which must have 2^k states.