Discrete Maths - 2017 Exam

Q1 – (a)

f : [-2, 2] → [-15, 1]
f(x) = x^2+3x-10 for all x [-2,2]

Injective ?

No since for f to be injective each input must map to a **unique** output – if f(x) = f(y) => x = y
Whereas this is not the case.

- f(0) = -10
- f(-1) = -10

Surjective ?

No since for f to be surjective it must **map to all elements in the codomain** - [-15, 1]. This is not the case as there does not exist an x that maps to -15.

Q1 – (b)

Let Q denote the relation on the set Z of itegers, where integers x and y satisfy x Q y if and only if

x – y = (x – y)(x + 2y)

Reflexive?
For a relation to be reflexive, all x must satisfy x Q x.
This is true for Q since (x – x) will always be 0 and the product of 0 is always 0.

Symmetric?
For a relation to be symmetric, all x and y must satisfy x Q y = y Q x.
The relation does not hold for x Q y
The relation also does not hold for y Q x

Since both are false, the false relation holds and the relation Q is therefore transitive.

x Q y	y Q x	(x Q y) ^ (y Q x)	→	Symmetric
F	F	F		TRUE

Transitive?
For a relation to be transitive for all x, y and z if x Q y and y Q z **then x Q z**.

The relation does not hold for x Q y
The relation also does not hold for y Q z
Therefore the relation for x Q z also does not hold

x Q y	y Q z	(x Q y) ^ (y Q z)	→	Transitive
F	F	F		TRUE

Anti-Symmetric?
For a relation to be anti-symmetric, for all x and y if x Q y and y Q x then **x = y**.
This is true for Q since the only time x Q y = y Q x holds is in the case that x = y.

Equivalence Relation?
For a relation to be an equivalence relation, the relation must be **reflexive, symmetric and transitive**. As shown above, this is not the case therefore the relation Q is not an equivalence relation.

Partial Order?
For a relation to be a partial order, the relation must be **reflexive, anti-symmetric and transitive**. As shown above, the relation Q is not transitive and is therefore not a partial order.

<u>Q2</u>

Let A = { 3^p | p ~ Z } with the operation of multiplication.

a) Is (A, .) a Semigroup ?

*A Semigroup is a set endowed with an associative binary operation***.**

Since multiplication is an associative binary operation, the set A is therefore a semigroup.

b) Is (A, .) a Monoid ?

*A monoid is a set endowed with an associative binary operation * that has an identity element e.*

*An identity element is an element e for the binary operation * s.t*

*e * x = x * e = x*

The set A has an identity element 1, so that for all x * 1 = x. Therefore the set A can be considered a monoid since it *is a semigroup, and it contains an identity element e*.

c) Is (A, .) a Group ?

A group is a set in which every element is invertable.

Every element of A is invertable through logarithmic such as $(3^p)^{-1}$ = log 3p. Therefore the set A can be considered to be a Group defined under the notation:

(A, *, 1)

d) Is A finite, countably infitite or uncountably infinite ?

The set A is definitely not finite as it is defined under the set of Integers (Z) which in themselves are infinite.

The set A can be considered countably infinite as it can be shown that all of the elements of A can be represented as a 1-1 correspondence with the set of Natural Numbers (N). Even negative rationals produced as a result of p being < 1 can be represented as some form of the natural numbers. This can be shown using Cantor's Zig Zag as found here https://bit.ly/2H6q7CV.

Q3

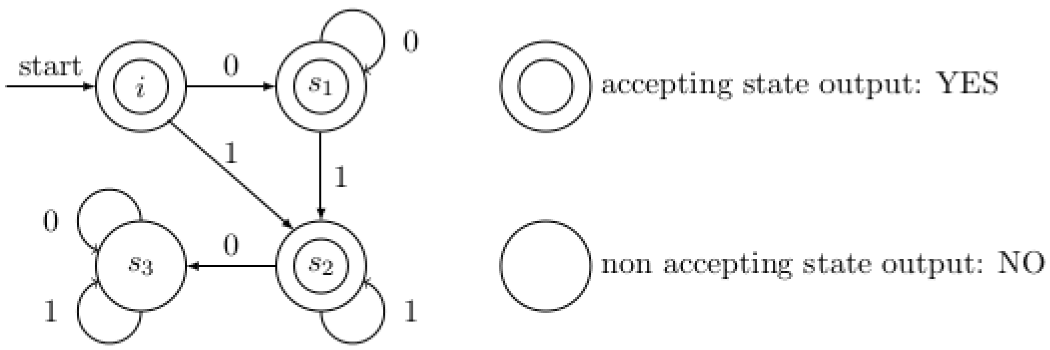
Let L be the language over the alphabet A ={ a, l, p } consisting of all words containing at least one of the substrings ala or pap.

a) Draw a Finite State Acceptor for the language L.

-Define States, Starting States, Finishing States, And Transition Table (See Below)

Example: Build a deterministic finite state acceptor for the regular language $L = \{0^m 1^n \mid m, n \in \mathbb{N}, m \geq 0, n \geq 0\}$

Accepting states in this examples: i, s_1, s_2
Non accepting states: s_3
Start states: i
Here $S = \{i, s_1, s_2, s_3\}$ $F = \{i, s_1, s_2\}$ $A = \{0,1\}$ $t : S \times A \to S$ $t(i,0) = s_1$ $t(i,1) = s_2$ $t(s_1,0) = s_1$ $t(s_1,1) = s_2$ $t(s_2,0) = s_3$ $t(s_2,1) = s_2$
Let's process some strings:



String	ε (empty string)
State (i)	i
Output	YES

String	0	0	1	1	1
State i	s_1	s_1	s_2	s_2	s_2
Output	YES				

String	1	1
State i	s_2	s_2
Output	YES	

String	1
State i	s_2
Output	YES

String	0	1	0	1
State i	s_1	s_2	s_3	s_3
Output	NO			

b) Devise a regular grammar in normal form that generates the language L. Specify the start symbol, the non-terminals and all the production rules.

A context-free grammar (V, A, <s>, P) is called a regular grammar if every production rule in P is of one of the three forms:
(i) <A>→ b<B>
(ii) <A>→ b
(iii) <A>→ ε

Start Symbol - <s>
Non-Terminals - <A> and <B>

Production Rules

1. <S> → a<A>
2. <S> → p<B>
3. <S> → empty worod
4. <A> → l<S>
5. <A> → empty word
6. <B> → a<S>
7. <B> → empty word

c) Write down a regular expression that gives the language L and justify your answer.

A = { a , l, p }

$L = A^* \circ [\ ((ala)^+ \ | (pap)^*) \ | ((ala)^* \ | (pap)^+)\ ] \circ A^*$

_ being concatenation

d) Consider the language L' over the alphabet A = { a, l, p } consisting of all words of the form $a^m l^{2m} p^m$ for m ∈ N*. Use the pumping lemma to show the language L' is not regular.

*If L is a regular language, then there is a number **p** (the pumping length) where if **w** is any word in **L** of length at least **p**,t hen **w** = **xuy** for words **x**, **y**, and **u** satisfying:*

Assume L is regular. Choose a string in terms of P that is easy to analyse. Let this string be:

$w = \boldsymbol{a^P l^{2P} p^P}$

Now break w into three components, x, u, and y.

If $|xu| \leq p$, then xu must consist of all a's as the first P characters in w. Also if $|u| > 0$, we must conclude $u = a^k$ for some k 0.

Therefore by the pumping lemma for all n, $\boldsymbol{xu^n y}$. By choosing n =2, we obtain the string $\boldsymbol{a^q l^{2P} p^P}$ with q>P. This string cannot be in L. Therefore we have found a contradiction and L is not regular.

1. $|u| > 0$, the length of u is positive;
2. $|xu| \leq p$;
3. $xu^n y \in L\ \forall\ n \geq 0$.

Q4

a)

i. ***What is meant by saying that a graph is complete?***

   A complete graph is a graph containing the highest number of edges possible for its given set of vertices.

ii. ***What is meant by saying a graph is regular?***

   A regular graph is a graph where every vertex has the same degree. A graph is considered to be k-regular if every vertex has a degree of k.

iii. ***What is the formula that relates the number of vertices of a tree with its number of edges?***

   The formula is as follows:

   #V = 2 * #E

iv. ***What is a spanning tree of a graph (V, E)?***

   A spanning tree of a graph (V, E) is a subgraph that is a tree which includes all of the vertices of V with the minimum possible number of edges.

b) Let (V, E) be the graph with

V = a, b, c, d, e, f and g.
E = ab, bc, cd, de, ae, cf, dg

i. ***Write down it's incidence table and its adjacency table***

   Incidence Table

	Ab	Bc	Cd	De	Ae	Cf	dg
a	1				1		
b	1	1					
c		1	1			1	
d			1	1			1
e				1	1		
f						1	

g							1

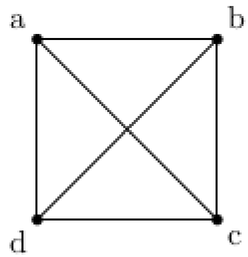
Adjacency Table

	a	b	c	d	e	f	g
a	0	1			1		
b	1	0	1				
c		1	0	1		1	
d			1	0	1		1
e	1			1	0		
f			1			0	
g				1			0

ii. Is this graph connected? Justify your answer

A connected graph is a graph where there exists a path in the graph from every u to every v.

The given graph is not connected as there does not exist a direct path between every vertice. For example there does not exist a path between the vertex d and the vertex b.



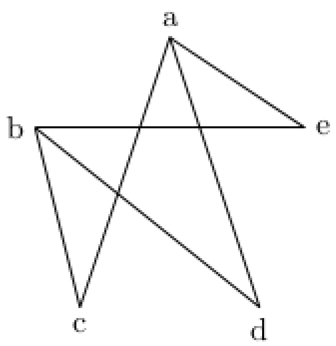
iii. Is this graph bipartite?

A bipartite graph is a graph that contains two seperate unique subsets of vertices that when taking every edge into account, every edge will be of the form E = vw where v is from subset one and w is from subset 2.

Example:

$V_1 = \{a, b\}$
$V_2 = \{c, d, e\}$
$V = \{a, b, c, d, e\}$
$E = \{ac, ad, ae, bc, bd, be\}$
is a complete bipartite graph.

iv. Does this graph have an Eulerian Trail?



An Eulerian trail in a graph is a trail that traverses every edge of that graph. In other words, an Eulerian trail is a walk that traverses every edge of the graph exactly once.

Trail ⇒ an edge is traversed at most once.
Eulerian ⇒ every edge is traversed.

v. Does this graph have a Hamiltonian Circuit

A Hamiltonian circuit in a graph is a simple circuit that passes through every vertex of the graph.

vi. Is this graph a tree?

Let (V, E) be a tree, then **#(E) = #(V) − 1**, where #(E) is the number of edges of the tree and #(V) is the number of vertices.