<u>Discrete Maths – Assignments Solutions</u>

<u>Assignment 1</u>

1) For any sets A and B, define A Δ B, the symmetric difference of A and B to be the set (A − B) ∪ (B − A). Prove that intersection ∩ distributes over Δ : A ∩ (B Δ C) = (A ∩ B) Δ (A ∩ C) for all sets A, B, and C using the proof methods employed in lecture. Venn diagrams, truth tables, or diagrams for simplifying statements in Boolean algebra such as Veitch diagrams are NOT acceptable and will not be awarded any credit.

$$A \cap (B \,\Delta\, C) = A \cap ((B - C) \cup (C - B))$$
$$= [A \cap (B - C)] \cup [A \cap (C - B)]$$

$$(A \cap B) \,\Delta\, (A \cap C) = [(A \cap B) - (A \cap C)] \cup [(A \cap C) - (A \cap B)]$$
$$= [A \cap (B - C)] \cup [A \cap (C - B)]$$

Q.E.D

2) Let R be the set of real numbers. For x, y ∈ R, x ∼ y iff x − y ∈ Q, i.e., if the difference x − y is a rational number. Determine whether or not the relation ∼ is :

a) Reflexive

b) Symmetric

c) Anti-Symmetric

d) Transitive

e) An Equivalence Relation

f) A Partial Order

a) If ∼ is a reflexive relation that implies for all x ∈ *R*, x ∼ *x*. In other words meaning that for all x ∈ *R*, *x* – *x* ∈ *Q*. This is true since x – x = 0 ∈ Q.

b) If ∼ is a symmetric relation that implies for all x,y ∈ *R*, x ∼ *y* = *y* ∼ *x*. This true since x – y = q ∈ *Q* and since y – x = -q ∈ *Q*.

c) If ∼ is an anti-symmetric relation that implies for all x,y ∈ *R*, if x ∼ *y* and y ∼ *x* then y = x. This is not true, take for example x = 1 and y = 0, x ∼ *y* = x – y = 1 – 0 = 1 ∈ *Q*. However, y ∼ *x* = 0 – 1 = - 1 ∈ *Q*. Clearly x != y.

d) If $\sim$ is a transitive relation that implies for all x,y,z $\in R$, if x $\sim y$ and y $\sim z$ then x $\sim z$. This is true, because for any x,y,z $\in R$ if x – y = p and y – z = q and p,q $\in Q$ then x – z = (x – y) - (x – z) = p – q $\in Q$.

e) $\sim$ is an equivalence relation because it is reflexive, symmetric and transitive.

f) $\sim$ is not a partial order as it fails to be anti-symmetric as shown above.

6) Let $f : [-2, 0] \rightarrow [0, 1]$ be the function defined by $f(x) = 1 / x^2 + 6x + 9$ for all $x \in [-2, 0]$.

Determine whether or not this function is injective and whether or not it is surjective. Justify your answers. Recall that $[-2, 0]$ is the set of all real numbers between -2 and 0 with the end-points of -2 and 0 included in the set.

a) f is an <u>injective</u> function as there does not exist an x, y $\in$ [-2, 0] s.t f(x) = f(y) = p. Each input in the domain maps to a unique output in the codomain.

b) f is not a <u>surjective</u> function as for all x $\in$ [-2, 0], y $\in$ [0, 1] there does not exist an f(x) = y. For example there does not exist an x $\in$ [-2, 0] s.t f(x) = 0.

7) Let $A = \{(x, y) \in R^2 \mid 2x - 3y = 0\}$ with the operation of addition given by

$$(x_1, y_1) + (x_2, y_2) = (x_1 + x_2, y_1 + y_2)$$

a) Is (A, +) a semigroup? Justify your answer.

b) Is (A, +) a monoid? Justify your answer.

c) Is (A, +) a group? Justify your answer.

a) *(A, +)* is a semigroup as it is a set endowed with an associative binary operation (addition) since $(x_1, y_1) + (x_2, y_2) = (x_2, y_2) + (x_1, y_1)$. Also, this can be associated over 2x – 3y = 0. Namely, $2(x_1, x_2) + 3(y_1, y_2) = 0$.

b) *(A, +)* is a monoid as it has an associated identity element which is $(x_1, y_1) = (0, 0)$. This is an identity element as for any $(x_1, y_1) \in R^2$ $(x_1, y_1) + (0, 0) = (x_1, y_1)$.

c) *(A, +)* is a group since for all $(x_1, y_1) \in R^2$ there exists an $(x_1, y_1)^{-1}$ namely in the form of

$(-x_1, -y_1)$.

Assignment 2

Question 1

a) Describe the formal language over the alphabet {a, b, c} generated by the context-free grammar whose non-terminals are S and A , whose start symbol is S , and whose production rules are the following:

(1) <S> → a <A>

(2) <A> → b <S>

(3) <S> → c

L = { $(ab)^n c \mid n \in N,\ n \geq 0$ }

Words generated under L are of the following form:

- $(ab)^n$ <S>
- $(ab)^n$ a <A>
- $(ab)^n$ c

b) Is this grammar regular? Justify your answer.

Yes, this grammar is regular as all of the above production rules satisfy the criteria necessary for a grammar to be regular. In other words, all of the production rules are of the form:

a) <A> ⟶ b <B>
b) <A> ⟶ b
c) <A> ⟶ ε

From our production rules given above, (1) is of type (a), (2) is of type (a) and (3) is of type (b) thus satisfying that it is a regular grammar.

c) Is this grammar innormal form? If it is not in normal form, then modify it to make it be in normal form. Explain why it generates the same language after it has been modified.

No, this grammar is not in normal form. To modify it to be in normal form the grammar would have to look like this:

L = { $(ab)^n c \mid n \in N,\ n \geq 0$ }

1) *<S> → a <A>*
2) *<A> → b <S>*
3) *<S> ⟶ c <B>*
4) *<B> ⟶ ε*

As before, words generated under this grammar are of the following form:

- $(ab)^n$ <S>
- $(ab)^n$ a <A>
- $(ab)^n$ c <B>
- $(ab)^n$ c

d) Write down a regular expression that gives the language from part (a) and justify your answer.

This can be generated under the following regular expression:

(ab)*c

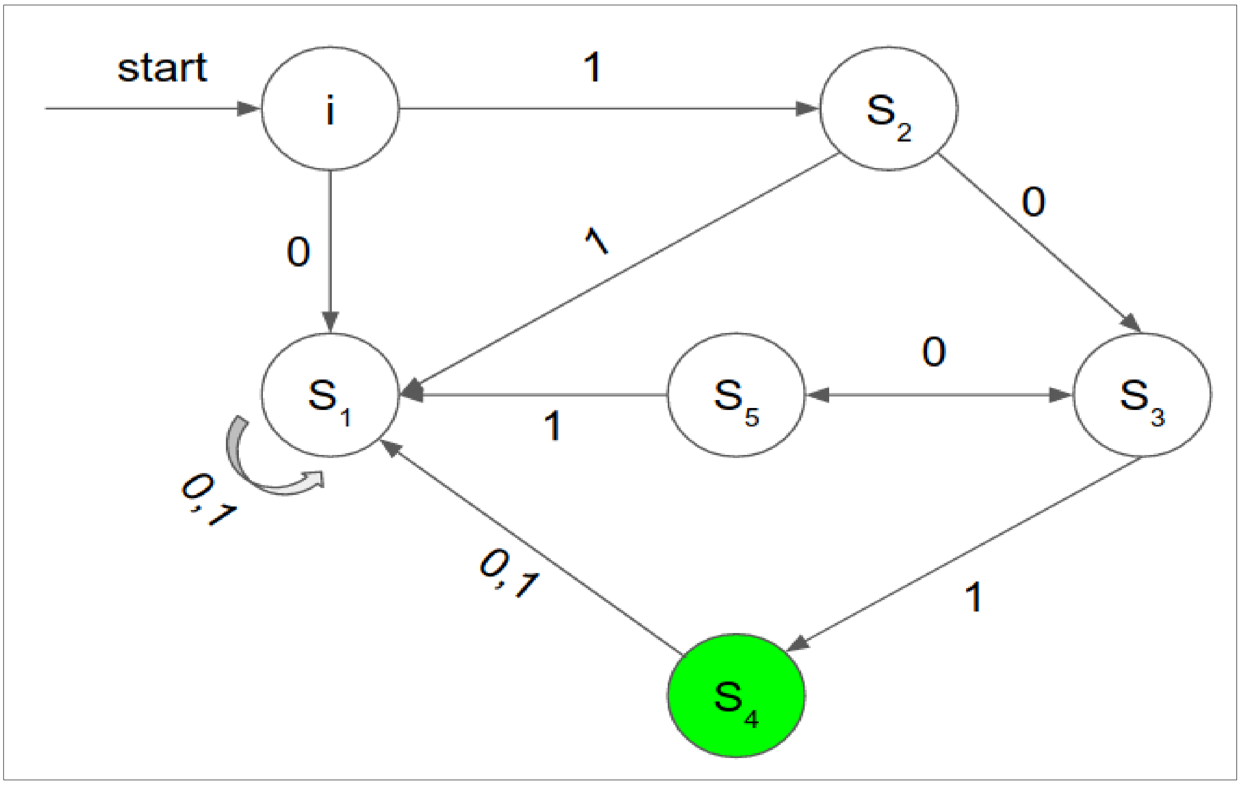
This expression gives the language K = { $(ab)^n$ c : n ≥ 0 } which is exactly our language L.

Question 2

Let L be the language consisting of all the strings of the form $10^{2m+1}1$ for m a natural number, m ≥ 0, i.e.

{101, 10001, 1000001, 100000001, . . . }

a) Draw a deterministic finite state acceptor that accepts the language L. Carefully label all the states including the start state and the finishing states as well as all the transitions. Make sure you justify it accepts all strings in the language L and no others.



A finite state acceptor (S, A, i, t, F) consists of:

- A finite set S of states
- A finite set A that is the input alphabet
- A starting state i ∈ S
- A transition mapping t : S × A → S
- A set F of finishing states, where F ⊆ S

- Accepting states: S_4
- Non accepting states: I, S_1 , S_2 , S_3 , S_5
- Start states: I

- S = {i, S_1 , S_2 , S_3 , S_4 , S_5 }
- F = {S_4}
- A = {0, 1}
- t : S × A →S

- $t(i, 0) = S_1$
- $t(i, 1) = S_2$

- $t(S_1 , 0) = S_1$
- $t(S_1 , 1) = S_1$

- $t(S_2 , 0) = S_3$
- $t(S_2 , 1) = S_1$
-
- $t(S_3 , 0) = S_5$
- $t(S_3 , 1) = S_4$

- $t(S_4 , 0) = S_1$
- $t(S_4 , 1) = S_1$

- $t(S_5 , 0) = S_3$
- $t(S_5 , 1) = S_1$

It can be shown that this FSA only accepts all strings in the language L and no others as if the string does not start with 1 it immediately transitions to an endless reject state (S_1).

States S_2 , S_3 , and S_5 ensure that there are 2m + 1 zero's following from the previous 1 value. If this is not the case, the FSA will transition once again to the endless reject state (S_1).

Test 10001:

String	1	0	0	0	1
State	S_2	S_3	S_5	S_3	S_4
Output	NO	NO	NO	NO	YES

Test 01011:

String	0	1	0	1	1
State	S_1	S_1	S_1	S_1	S_1
Output	NO	NO	NO	NO	NO

It can be therefore taken that the Finite State Acceptor as shown above accepts all strings in the language L and no other strings.

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

$L = \{10^{2m+1}1 \mid m \text{ a natural number}, m \geq 0\}$

{101, 10001, 1000001, 100000001, . . . }

Start Symbol = <S>

Non-Terminals = <A>, <B>, <C>, <D>, <E>

Production Rules:

1. <S> → 1 <A>
2. <A> → 0 <B>
3. <A> → 0 <D>
4. <B> → 1 <C>
5. <C> → ε
6. <D> → 0 <E>
7. <E> → 0 <A>
8. <E> → 0 <B>

101	10001	10000001
1. → 1	1. → 1	1. → 1
2. → 10	3. → 10	3. → 10
4. → 101	6. → 100	6. → 100
5. → 101ε	8. → 1000	7. → 1000
	4. → 10001	3. → 10000
	5. → 10001ε	6. → 100000
		8. → 1000000
		4. → 10000001
		5. → 10000001ε

Question 3

Let M be the language

{0101, 001001, 00010001, 0000100001, . . . }

$M = \{ (0^m 1)^m \mid m \in N,\ m \geq 1 \}$

(a) Use the Pumping Lemma to show this language is not regular.

<u>The Pumping Lemma</u>

If A is a Regular Language, then A has a Pumping Length ‘P’ such that any string ‘S’ where $|S| \geq P$ may be divided into 3 parts S = xyz such that the following conditions are true

1. $x\,y^i\,z \in A$ for every $i \geq 0$
2. $|y| \geq 0$
3. $|xy| < P$

<u>To prove that a language is NOT REGULAR using the Pumping Lemma:</u>

1. Assume that A is regular
2. A must then have a Pumping Length P
3. All strings longer than P can be pumped $|S| \geq P$
4. Now find a string ‘S’ in A such that $|S| \geq P$
5. Divide S into x y and z
6. Show that $x\,y^i\,z \,!\!\in A$ for some i
7. Then consider all ways that S can be divided into x y and z
8. Show that none of these satisfy all 3 pumping conditions at the same time
9. S cannot be Pumped == CONTRADICTION

<u>Proof:</u>

Assume M is a regular language, therefore it must have a pumping length p such that a string $S = (0^P 1)^P$ and S is a word under the language M.

1. Divide such a string S int [x] [y] and [z]
2. Let the pumping length P = 7

   Therefore, S = 0000000100000001

3. Explore the different possibilities of breaking up S into [x] [y] and [z]

Case 1 (The y is right of 1):

$$\overset{x}{[00000001]}\ \overset{y}{[0000]}\ \overset{z}{[0001]}$$

Case 2 (The y contains 1):

$$\overset{x}{[0000000]}\ \overset{y}{[1000]}\ \overset{z}{[00001]}$$

Case 3 (The y is left of 1):

$$\overset{x}{[000]}\ \overset{y}{[0000]}\ \overset{z}{[100000001]}$$

Show that x y^i z !∈ M for some I
Let i = 2, therefore test S = x y^2 z

Case 1: S = $\overset{x}{[00000001]}\overset{y^2}{[00000000]}\overset{z}{[0001]}$

This S !∈ M since 7 x 0's and 1 x 1 followed by 11 x 0's and 1 x 1 which does not satisfy *M =* $\{ (0^m1)^m \}$ and therefore S !∈ M.

Case 2: S = $\overset{x}{[0000000]}\overset{y^2}{[10001000]}\overset{z}{[00001]}$

This S !∈ M since 7 x 0's and followed by $(1000)^2$ and then further followed by 0001 which does not satisfy M = $\{ (0^m1)^m \}$ and therefore S !∈ M.

Case 3: S = $\overset{x}{[000]}\overset{y^2}{[00000000]}\overset{z}{[100000001]}$

This S !∈ M since 11 x 0's and 1 x 1 followed by 7 x 0's and 1 x 1 which does not satisfy M = $\{ (0^m1)^m \}$ and therefore S !∈ M.

****REVERT BACK TO CONDITIONS 1-3 AND SHOW HOW THEY ARE NOT PROVED****

Condition 1 proven false above.

Condition 2 is true.

Condition 3 - $|xy| < P$ – is false in all of the above cases.

Therefore using the Pumping Lemma it is not possible for all strings under L to satisfy the conditions of the Pumping Lemma, therefore the language L is not regular.

(a) Write down the production rules of a context-free grammar that generates exactly M.

$M = \{ (0^m1)^m \mid m \in N,\ m \geq 1 \}$

{0101, 001001, 00010001, 0000100001, . . . }

<u>Production Rules:</u>

1. <S> → 0 <A> 01
2. <A> → 0 <A> 0
3. <A> → 1

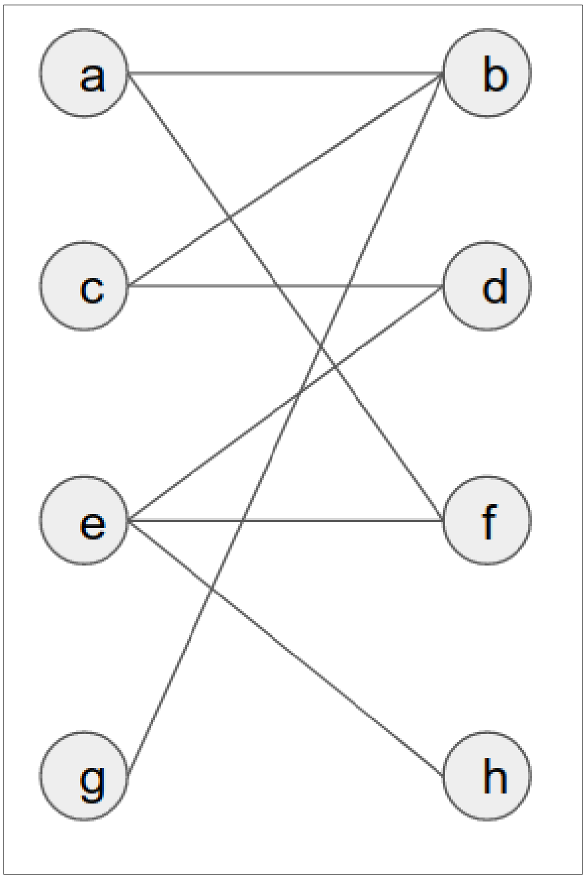
All words generated under this grammar are of the following form:

- 0 <A> 01
- 0^n <A> 0^n1
- 0^n10^n1

Question 4

Let (V, E) be the graph with vertices a, b, c, d, e, f, g, and h, and edges ab, bc, cd, de, ef, af, bg, and eh.

a) Draw this graph.



b) Write down the graphs incidence table & incidence matrix

	AB	BC	CD	DE	EF	AF	BG	EH
A	1					1		
B	1	1					1	
C		1	1					
D			1	1				
E				1	1			1
F					1	1		
G							1	
H								1

c) Write down the graphs adjacency table & adjacency matrix

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

d) Is this graph complete?

No, this graph is not complete. As for in order for a graph to be complete there must exist the maximum number of edges possible for the given vertices. This is not the case for the graph (V, E) for example there does not exist the edge fg.

e) Is this graph bipartite?

Yes this graph is bipartite. As for in order for a graph to be bipartite there must exist two distjoint subsets A = (V_1), B = (V_2) such that A U B = (V) and that for every edge ab an element of (E), a must be an element of V_1 and b must be an element of V_2.

Let A = (V_1 , E_1)
V_1 = { a, c, e, g}

Let B = (V_2 , E_2)
V_2 = { b, d, f, h}

Take for example the edge ef, e $\in$ V_1 and f $\in$ V_2.