Q1:

1) (10 points)

(a) Describe the formal language over the alphabet $\{a, b, c\}$ generated by the context-free grammar whose non-terminals are $\langle S\rangle$ and $\langle A\rangle$, whose start symbol is $\langle S\rangle$, and whose production rules are the following:
(1) $\langle S\rangle \to a\langle S\rangle a$
(2) $\langle S\rangle \to b\langle A\rangle$
(3) $\langle A\rangle \to b\langle A\rangle b$
(4) $\langle A\rangle \to c\langle A\rangle$
(5) $\langle A\rangle \to c$
In other words, describe the structure of the strings generated by this grammar.

(b) Use the Pumping Lemma to prove that the language from part (a) is not regular.

(1) $\langle S\rangle \xrightarrow{(1)\times x} a^x\langle S\rangle a^x \xrightarrow{(2)} a^x b\langle A\rangle a^x \xrightarrow{(3)\times y} a^x b b^y\langle A\rangle b^y a^x \xrightarrow{(4)\times z} a^x b b^y c^z\langle A\rangle b^y a^x \xrightarrow{(5)} a^x b b^y c^z c\, b^y a^x$

$\langle S\rangle \xrightarrow{(1)\times x} a^x\langle S\rangle a^x \xrightarrow{(2)} a^x b\langle A\rangle a^x \xrightarrow{(4)\times z} a^x b\, c^z\langle A\rangle a^x \xrightarrow{(5)} a^x b c^z c\, a^x$

$\langle S\rangle \xrightarrow{(1)\times x} a^x\langle S\rangle a^x \xrightarrow{(2)} a^x b\langle A\rangle a^x \xrightarrow{(5)} a^x b c a^x$

$L = \{a^x b b^y c^z c\, b^y a^x \mid x, y, z \subseteq N\} \cup \{a^x b c^z c\, a^x \mid x, z \subseteq N\} \cup \{a^x b c a^x \mid x \subseteq N\}$

(2) Introduction of pumping lemma:

If L is a regular language, $\exists p \in N$ such that $\forall w \in L$

If $|w| \geq p$, we have $x\,u\,y$

- $u \neq \epsilon$, $|u| \geq 1$
- $|xu| \leq p$ (first two concatenation has to less than p) $|u| = n_1$
- $x u^n y \in L$, $\forall n \geq 0$

Assume $a^x b b^y c^z c b^y a^x$

Let $w_1 = a^n b b^n c^n c b^n a^n$, then $x_1 \in L$ and $|w_1| = 5^n + 2$

Let $|x| = n_1$ $|u| = n_2$

$n_1 + n_2 \leq p \Rightarrow a^{n_1+n_2}$

$w_1 = a^{n_1+n_2} y = a^n b b^n c^n c b^n a^n$

Hence $y = a^{n-(n_1+n_2)} b b^n c^n c b^n a^n$

$= a^{n_3} b b^n c^n c b^n a^n$ $[p = n_1+n_2+n_3]$

$x\,u\,y = a^{n_1} a^{n_2} a^{n_3} b b^n c^n c b^n a^n$

$x\,u^2 y = a^{n_1} a^{n_1} a^{n_2} a^{n_3} b b^n c^n c b^n a^n$

$= a^{n_1+n_2+n_3} a^{n_1} b b^n c^n c b^n a^n$

Assume $a^x b c^z c a^x$

Let $w_2 = a^n b c^n c a^n$, then $x_2 \in L$ and $|w_2| = 3^n + 2$

Let $|x| = n_1$ $|u| = n_2$

$n_1 + n_2 \leq p \Rightarrow a^{n_1+n_2}$

$w_2 = a^{n_1+n_2} y = a^n b c^n c a^n$

Hence $y = a^{n-(n_1+n_2)} b c^n c a^n$

$= a^{n_3} b c^n c a^n$ $[p = n_1+n_2+n_3]$

$x\,u\,y = a^{n_1} a^{n_2} a^{n_3} b c^n c a^n$

$x\,u^2 y = a^{n_1} a^{n_2} a^{n_3} a^{n_2} b c^n c a^n$

$= a^{n_1+n_2+n_3} a^{n_2} b c^n c a^n$

$= a^{p+n_2} b b^n c^n c b^n a^n$

Since $n_2 \geq 1$, $p+n_2 > p \notin L$

so W is not regular

$= a^{p+n_2} b c^n c a^n$

Since $n_2 \geq 1$, $p+n_2 > p \notin L$

so W is not regular

Assume $a^x bc a^x$

Let $w_3 = a^n bc a^n$, then $x \in L$ and $|w_3| = 2^n + 2$

if $|x_1| = n_1$, $|u| = n_2$

$n_1 + n_2 \leq p \Rightarrow a^{n_1+n_2}$

$w = a^{n_1+n_2} y = a^n bc a^n$

Hence $y = a^{n-(n_1+n_2)} bc a^n$

$= a^{n_3} bc a^n \quad [p = n_1+n_2+n_3]$

$x u y = a^{n_1} a^{n_2} a^{n_3} bc a^n$

$x u^2 y = a^{n_1+n_2+n_3} a^{n_2} bc a^n$

$= a^{p+n_2} bc a^n$

Since $n_2 \geq 1$, $p+n_2 > p \notin L$

Q2:

2) (20 points) Let L be the language consisting of all binary numbers divisible by 8. Note that any binary number starting with 0 and containing more than one symbol is considered improper and should be rejected.

(a) Draw a deterministic finite state acceptor that accepts the language L. Carefully label all the states including the starting 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.

(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. Make sure you justify it generates all strings in the language L and no others.

(c) Prove by applying the definition of a regular language that the language L is regular.

(d) Write down a regular expression that gives L and justify your answer.

Let the alphabet $A = \{0, 1\}$

(a) 8: 00|000

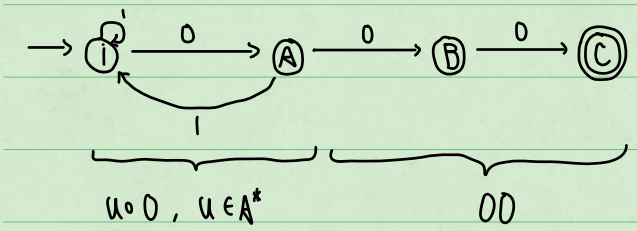
16: 0|0000

24: 011|000

32: |00 000

Hence $L = \{w \in A^* | w = u\,0\,000,\ u \in A^*\}$

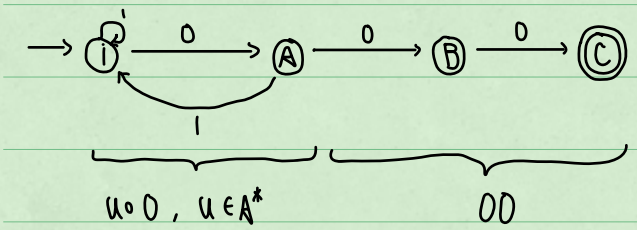
Hence any binary number whose last 3 digit are 0 is divisible by 8



(b) $\langle X \rangle \longrightarrow a\langle Y \rangle$, $a \in A^*$

$\langle X \rangle \longrightarrow \epsilon$

1. $\langle S \rangle \longrightarrow 1\langle S \rangle$
2. $\langle S \rangle \longrightarrow 0\langle A \rangle$
3. $\langle A \rangle \longrightarrow 1\langle S \rangle$
4. $\langle A \rangle \longrightarrow 0\langle B \rangle$
5. $\langle B \rangle \longrightarrow 0\langle C \rangle$
6. $\langle C \rangle \longrightarrow \epsilon$



(C) Recall that the definition of a regular language allows for finite subset of A^*, the Kleene star, concatenations, and unions. Note that

$$L = \{w \in A^* \mid w = u \circ 000,\ u \in A^*\}$$

Therefore, we can let $L_1 = \{000\}$ be the language consisting of just the string 000 of interest

L_1 is a finite set, so it is allowed in the definition of a regular language

Let $L_2 = \{0,1\}$. L_2 is finite, hence likewise allowed

Let $L_3 = L_2^*$, the Kleene star applied to L_2. The language $L_3 = A^*$, i.e. it is the set of all words that can be formed over the alphabet $A = \{0,1\}$. Set $L_4 = L_3 \circ L_1$

Note that the words in L_4 have exactly the structure of the words in L, and in fact, $L = L_4$.

Note also that the solution here is by no means unique. No two of you will necessarily have arrived at the same exact expression, order of labelling of the intermediate languages L_i that come into the definition of a regular language as applied to L

(d) create a regular expression that gives L

$L = A^* \circ 000$

Q3:

3) (10 points) Let A be a finite alphabet.
(a) Let L be a regular language over the alphabet A. Prove that $A^* \setminus L$, the complement of L in A^*, is also a regular language. (Hint: Think about the equivalent conditions characterising a regular language and figure out which one is easiest to check here.)
(b) Let L_1 and L_2 be regular languages over the alphabet A. Prove that their intersection $L_1 \cap L_2$ is a regular language. (Hint: Use part (a) and de Morgan's.)

According to the theorem, we can easily answer these two questions above

Theorem: The collection of regular languages L is also closed under the following two operations:

(a) Complement, i.e. if L is a regular language (i.e. $L \in C$), then $A^* \setminus L$ is a regular language.

Hence we proved question a

(b) Intersection, i.e. if L', L'' are regular languages (i.e. $L' \in C$ and $L'' \in C$), then their intersection $L' \cap L''$ is a regular language

Hence we proved question b

Another way to prove:

(a) we have two elements which are L and A^*/L (A^*: the set of all words in the alphabet)

which means $(L) \cup (A^* \setminus L) = A^*$

theorem: If L is a regular language, then it's complement $\bar{L}$ is also regular

Proof: If L is regular, there is a DFA M that decides L. Construct a new mechaine $\bar{M}$ from M by changing every non-accepting state in M to an accepting state and every accepting state in M to a non-accepting state. For any input string $x_0 \in L$, x_0 drives M to an accepting state. That same x_0 will drive $\bar{M}$ to a non-accepting state. Likewise, input strings x_0 that are not in L will drive M to a non-accepting state, which means that same x_0 drives $\bar{M}$ to an accepting state. Thus, $\bar{M}$ will accept x_0 if and only if $x_0 \in \bar{L}$

(b) we already know from question (a) that a regular language's complement is still a regular language

And we got the demorgan's law : $\overline{A \cup B} = \bar{A} \cap \bar{B}$, $\overline{A \cap B} = \bar{A} \cup \bar{B}$

Set $A = L_1$ $B = L_2$ A and B are both regular languages

$\bar{A}$ and $\bar{B}$ are still regular languages according to (a)

$\bar{A} \cup \bar{B}$ is regular language according to (a)

$\bar{A} \cup \bar{B} = \overline{A \cap B}$ according to demorgan's law set $A \cap B = C$

$\overline{A \cap B} = \bar{C}$, $\bar{C}$ is a regular language , so $C = A \cap B$ is a regular language too

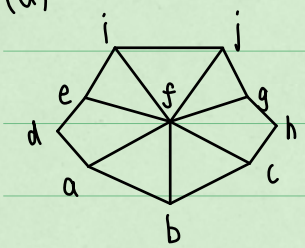
according to (a). Hence we proved L_1, L_2 are regular languages , then $L_1 \cap L_2$ is a regular language

Q4

4) (20 points) Let (V, E) be the graph with vertices a, b, c, d, e, f, g, h, i, and j, and edges ab, bc, af, bf, cf, ef, fg, fi, fj, ad, ch, de, gh, ei, ij, and gj.

(a) Draw this graph.
(b) Write down this graph's incidence table and its incidence matrix.
(c) Write down this graph's adjacency table and its adjacency matrix.
(d) Is this graph complete? Justify your answer.
(e) Is this graph bipartite? Justify your answer.
(f) Is this graph regular? Justify your answer.
(g) Does this graph have any regular subgraph? Justify your answer.
(h) Give an example of an isomorphism φ from the graph (V, E) to itself satisfying that $\varphi(b) = b$.
(i) Is the isomorphism from part (h) unique or can you find another isomorphism ψ that is distinct from φ but also satisfies that $\psi(b) = b$? Justify your answer.

(a)



(b)

	ab	bc	af	bf	cf	ef	fg	fi	fj	ad	ch	de	gh	ei	ij	gj
a	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0
b	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
c	0	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0
d	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
e	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0
f	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0
g	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1
h	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0
i	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0
j	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1

(c)

	a	b	c	d	e	f	g	h	i	j
a	0	1	0	1	0	1	0	0	0	0
b	1	0	1	0	0	1	0	0	0	0
c	0	1	0	0	0	1	0	1	0	0
d	1	0	0	0	1	0	0	0	0	0
e	0	0	0	1	0	1	0	0	1	0
f	1	1	1	0	1	0	1	0	1	1
g	0	0	0	0	0	1	0	1	0	1
h	0	0	1	0	0	0	1	0	0	0
i	0	0	0	0	1	1	0	0	0	1
j	0	0	0	0	0	1	1	0	1	0

(d) No as for example, there is no edge from d to f and h to f

i.e. $df \notin E$, $hf \notin E$

(e) No as the graph contains the complete subgraph, $V' = \{a, b, f\}$, $E' = \{ab, af, bf\}$ which can not be partitioned

(f) No, because the vertex f has 7 edges while vertex d has only 2 edges each vertex doesn't have same number of edges

(g) In graph theory, a regular graph is a graph where each vertex has the same number of neighbours

Vertex a has 3 neighbours
Vertex b has 3 neighbours
Vertex c has 3 neighbours
Vertex d has 2 neighbours
Vertex e has 3 neighbours
Vertex f has 7 neighbours
Vertex g has 3 neighbours
Vertex h has 2 neighbours
Vertex i has 3 neighbours
Vertex j has 3 neighbours

For vertex neighbours numbers greater and equal to 3 can try to form a regular subgraph

for vertex a, b, c, edge ac is missing

for vertex g, i, j and e, i, j, edge ig, je is missing

for vertex e, a, g, c, edge ae, eg, gc is missing

for vertex a, i, e, edge ai is missing

for vertex c, j, g, edge cj is missing

for vertex a, j, e, edge aj is missing

for vertex c, i, g, edge ci is missing

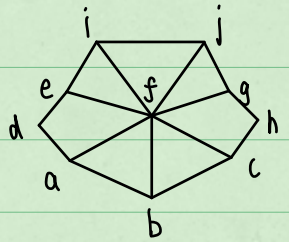
for vertex a, g, e, edge ag is missing

for vertex c, e, g, edge ce is missing

Hence this graph doesn't have any regular subgraph

(h) Example of a self isomorphism is

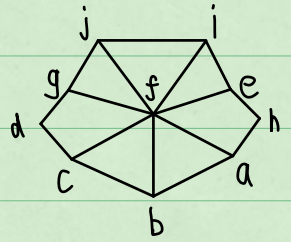
the identity map $\begin{cases} \varphi(a)=a \\ \varphi(b)=b \\ \varphi(c)=c \\ \varphi(d)=d \\ \varphi(e)=e \\ \varphi(f)=f \\ \varphi(g)=g \\ \varphi(h)=h \\ \varphi(i)=i \\ \varphi(j)=j \end{cases}$



(i) the isomorphism from part h is not unique

the identity map

$$\begin{cases} \varphi(a) = C \\ \varphi(b) = b \\ \varphi(c) = a \\ \varphi(d) = d \\ \varphi(e) = g \\ \varphi(f) = f \\ \varphi(g) = e \\ \varphi(h) = h \\ \varphi(i) = j \\ \varphi(j) = i \end{cases}$$



the identity map

$$\begin{cases} \varphi(a) = C \\ \varphi(b) = b \\ \varphi(c) = a \\ \varphi(d) = h \\ \varphi(e) = g \\ \varphi(f) = f \\ \varphi(g) = e \\ \varphi(h) = d \\ \varphi(i) = j \\ \varphi(j) = i \end{cases}$$