1. (a) Prove the following expressions by applying the rules in the table: [10 marks]

i. $((A \wedge B) \rightarrow C) \leftrightarrow (A \rightarrow (B \rightarrow C))$

ii. $((A \rightarrow C) \wedge (B \rightarrow C)) \leftrightarrow ((A \vee B) \rightarrow C)$

1	$\neg X \vee Y \leftrightarrow X \rightarrow Y$
2	$\neg(X \vee Y) \leftrightarrow \neg X \wedge \neg Y$
3	$\neg(X \wedge Y) \leftrightarrow \neg X \vee \neg Y$
4	$(X \wedge Y) \vee Z \leftrightarrow (X \vee Z) \wedge (Y \vee Z)$

I.

$(A \wedge B) \rightarrow C$	$\neg X \vee Y \leftrightarrow X \rightarrow Y$	(1)
$\neg(A \wedge B) \vee C$	$\neg(X \wedge Y) \leftrightarrow \neg X \vee \neg Y$	(3)
$(\neg A \vee \neg B) \vee C$	$(X \vee Y) \vee Z \leftrightarrow X \vee (Y \vee Z)$ (A rule in the table?)	
$\neg A \vee (\neg B \vee C)$	$\neg X \vee Y \leftrightarrow X \rightarrow Y$	(1)
$A \rightarrow (\neg B \vee C)$	$\neg X \vee Y \leftrightarrow X \rightarrow Y$	(1)
$A \rightarrow (B \rightarrow C)$		

II.

$(A \vee B) \rightarrow C$	$\neg X \vee Y \leftrightarrow X \rightarrow Y$	(1)
$\neg(A \vee B) \vee C$	$\neg(X \vee Y) \leftrightarrow \neg X \wedge \neg Y$	(2)
$(\neg A \wedge \neg B) \vee C$	$(X \wedge Y) \vee Z \leftrightarrow (X \vee Z) \wedge (Y \vee Z)$	(4)
$(\neg A \vee C) \wedge (\neg B \vee C)$	$\neg X \vee Y \leftrightarrow X \rightarrow Y$	(1)
$(A \rightarrow C) \wedge (B \rightarrow C)$		

(b) Are the following two expressions tautologous? Using a truth table justify your answer.

i. $(A \wedge (\neg B \rightarrow (B \rightarrow C))) \leftrightarrow A$

ii. $((A \rightarrow B) \wedge (B \rightarrow C)) \rightarrow (A \rightarrow C)$

[10 marks]

Yes

A	B	C	¬B	B → C	¬B → (B → C)	A ∧ (¬B → (B → C))	III
T	T	T	F	T	T	T	T
T	T	F	F	F	T	T	T
T	F	T	T	T	T	T	T
T	F	F	T	T	T	T	T
F	T	T	F	T	T	F	T
F	T	F	F	F	T	F	T
F	F	T	T	T	T	F	T
F	F	F	T	T	T	F	T

A	B	C	A → B	B → C	(A → B) ∧ (B → C)	A → C	IV
T	T	T	T	T	T	T	T
T	T	F	T	F	F	F	T
T	F	T	F	T	F	T	T
T	F	F	F	T	F	F	T
F	T	T	T	T	T	T	T
F	T	F	T	F	F	T	T
F	F	T	T	T	T	T	T
F	F	F	T	T	T	T	T

2. Having the following proof system, prove the argument [p∨q∨ z→r, r→p] ⊢ (q∨z)→r.

[10 marks]

Rule	Sequent	Supporting proofs
imm	$\mathcal{F} \vdash A$	$A \in \mathcal{F}$
and_intro	$\mathcal{F} \vdash A \wedge B$	$\mathcal{F} \vdash A, \mathcal{F} \vdash B$
and_elim	$\mathcal{F} \vdash C$	$A \wedge B \in \mathcal{F}, [A, B \mid \mathcal{F}] \vdash C$
imp_intro	$\mathcal{F} \vdash A \rightarrow B$	$[A \mid \mathcal{F}] \vdash B$
imp_elim	$\mathcal{F} \vdash B$	$A \rightarrow B \in \mathcal{F}, \mathcal{F} \vdash A$
or_intro_left	$\mathcal{F} \vdash A \vee B$	$\mathcal{F} \vdash A$
or_intro_right	$\mathcal{F} \vdash A \vee B$	$\mathcal{F} \vdash B$
or_elim	$\mathcal{F} \vdash C$	$A \vee B \in \mathcal{F}, [A \mid \mathcal{F}] \vdash C, [B \mid \mathcal{F}] \vdash C$

One derivation is:

[p∨(q∨ z)→r, r→p] ⊢ (q∨z)→r (imp-intro)

[p∨(q∨ z)→r, r→p, q∨z] ⊢ r (imp-elim)

  [r→p, (q∨z)] ⊢ p∨(q∨ z) (or_intro_left)

  [r→p, q∨z] ⊢ p (or_elim)

    [r→p, q] ⊢ p unsuccessful

    [r→p, z] ⊢ p unsuccessful

  [r→p, (q∨z)] ⊢ q∨ z (or_intro_right)

    [r→p, (q∨z)] ⊢ q∨ z (or_intro_left)

    [r→p, (q∨z)] ⊢ q (or_elim)

    [r→p, q] ⊢ q , [r→p, z] ⊢ q (or_elim)

    [r→p, q] ⊢ q (imm)

[p →q, q→r] ⊢ (p→s) (imp-intro)

    [p →q, q→r, p] ⊢ s ---

  [p →q, q→r] ⊢ p→r (or-intro- right)

    [p →q, q→r] ⊢ (p→r) (imp-intro)

    [p →q, q→r, p] ⊢ r (imp-elim)

    [p →q, p] ⊢ q (imp-elim)

    [p] ⊢ p (immidiate)

3. Assume we have the following set of premises:

   [¬c∧d, (b∧ e)→c, ¬a→e, ¬b→ ¬ (e∧d), e]

a) Convert the above expression into a Conjunctive Normal Form (CNF). [8 marks]

   [¬c ∧ d ∧ (¬b∨ ¬e∨ c) ∧ (a∨e) ∧ (b∨¬e∨¬d) ∧ e]

b) Convert the CNF into a clausal form. [2 marks]

   [[¬c], [d], [¬b, ¬e, c], [a, e], [b, ¬e, ¬d], [e]]

c) Using Resolution show whether ¬e can be proved from the premises or not. [10 marks]

Using the following resolution rules, after adding [¬a] to the expression, we show that [a] **cannot be proved** by the premises:

[[¬c], [d], [¬b, ¬e, c], [a, e], [b, ¬e, ¬d], [e], [¬a]]	([¬c], [¬b, ¬e, ¬d)
[[d], [¬b, ¬e], [a, e], [b, ¬e, ¬d], [e], [¬a]]	([d], [b, ¬e, ¬d])
[[¬b, ¬e, c], [a, e], [b, ¬e], [e], [¬a]]	([e],[b, ¬e])
[[¬b, ¬e, c], [a, e], [b], [¬a]]	([b],[¬b, ¬e, c])
[[¬e, c], [a, e], [¬a]]	([¬a],[a,e])
[[¬e, c], [e]]	
[[c]]	

N.B. There are other ways of applying the Resolution rules to reach to a non-empty clause.

4.

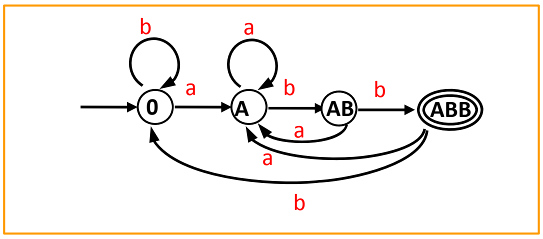
(a) Briefly describe one advantage of each of deterministic and non-deterministic FSA. [4 marks]

The determinism is important when **implementing a program or physical machine** for recognising a regular language because the machine itself does not make choices and it is determined to change the states as it supposed to.

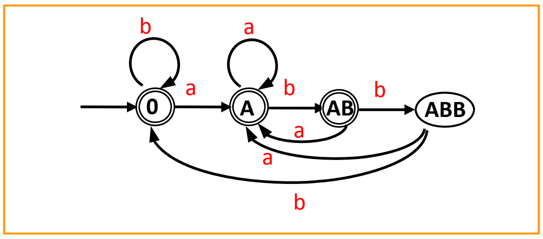
Non-deterministic FSMs are a lot **easier to build**, especially when we want to construct a FSM to accept a **relatively complicated class of languages, modelling languages/systems**, they provide a **more intuitive** framework than deterministic machines

(b) Consider the language L of strings containing substring "abb" over the alphabet {a,b}.

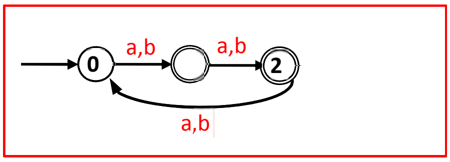
i. Draw a FSM that accepts L. [5 marks]



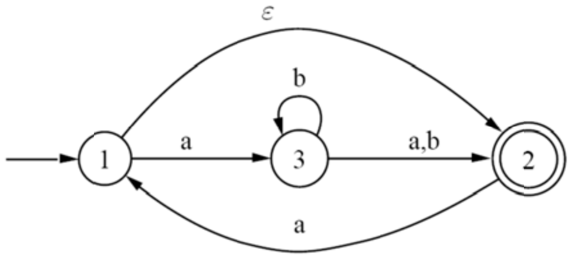
ii. Draw a FSM that accepts $\overline{L}$. ($\overline{L}$ is the complement of L) [5 marks]



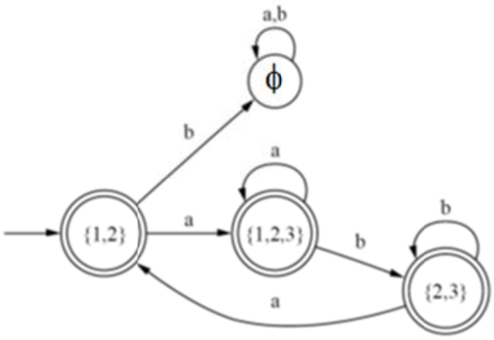
(c) Draw a FSM that accepts all string with the length K such that: K mod 3 ≠0. [4 marks]



(d) Convert the following non-deterministic FSM to a deterministic FSM. [10 marks]



Answer:



5. The following languages are defined over the alphabet {a, b}

a) Is the language L= $\{a^m b^n \mid m \text{ is even, } n \text{ is odd}\}$ a regular language? [2 marks]

Yes

b) Give a regular expression for each of the following languages.

i) L_1= {strings begin with aa} [2 marks]

aa(a|b)*

ii) L_2= {strings end with bb} [2 marks]

(a|b)*bb

iii) $L_1 \cap L_2$ (intersection of L_1 and L_2) [2 marks]

aa(a|b)*bb

iv) L_1L_2 (concatenation of L_1 and L_2) [to be deleted]

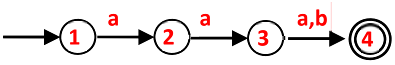
aa(a|b)*bb

v) L_3 that contains strings with odd length. [4 marks]

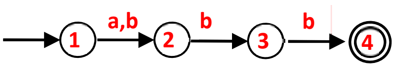
(a|b)((a|b)(a|b))*

c) Draw the following FSMs:

i. M_1 that accepts language L_1. [2 marks]

M_1: 

ii. M_2 that accepts language L_2. [2 marks]

M_2: 

iii. M_3 that accepts language $L_1 \cup L_2$, having a single final state. [6 marks]

M_3: