Q1. Logic

(a) Prove, or find a counterexample to, each of the following assertions:

(i) If $\alpha \models \gamma$ or $\beta \models \gamma$ (or both) then $(\alpha \land \beta) \models \gamma$

Assuming M a model such that M ent (a and b) (this means both a and b are true). since a ent g, if a is true in M then g is true in M. this means M ent g when a is true.
at the same time we can see it for b. in the case of (a and b) they're both true simultaneously. since either one of the two being true is sufficient to implies g is true, it follows that g must be true when both a and b are true.

(ii) If $(\alpha \land \beta) \models \gamma$ then $\alpha \models \gamma$ or $\beta \models \gamma$ (or both).

if a = p, b = q and g = (p and q) then ofc (p and q) ent (p and q), but it is not true that p ent (p and q) and equally it is not true that q ent (q and p).

(iii) If $\alpha \models (\beta \lor \gamma)$ then $\alpha \models \beta$ or $\alpha \models \gamma$ (or both).

assuming M a model such that M ent a. since a ent b or g, if a is true in M it is also true or b or g (or both) iin M. this means or b or g are true while a is true.
this implies that a ent b or a ent g (or both).

(b) Decide whether each of the following sentences is valid, unsatisfiable, or neither.

(i) $Smoke \implies Smoke$

!Smoke or Smoke -> tautology

(ii) $Smoke \implies Fire$

!Smoke or Fire -> neither

(iii) $(Smoke \implies Fire) \implies (\neg Smoke \implies \neg Fire)$

!(!Smoke or Fire) or (Smoke or !Fire) -> (Smoke and Fire) or (Smoke or !Fire) -> neither
S = 0, F = 0 : 1 ; S = 0, F = 1 : 0

(iv) $Smoke \lor Fire \lor \neg Fire$

tautology (Fire or !Fire)

(v) $((Smoke \land Heat) \implies Fire) \iff ((Smoke \implies Fire) \lor (Heat \implies Fire))$

(!(S and H) or F) iif ((!S or F) ir (!H or F)) -> (!S or !H or F) iif (!S or !H or F) we then note that the left part is the same as the right part then we can see it as Phi iif Phi -> (!Phi or Phi) and (Phi or !Phi) that's the same of doing (denoting (!Phi or Phi) as Lem) Lem and Lem. So studying !Phi or Phi we have:
(S and H and !F) or (!S or !H or F) the good fact about this is that we can study the only case when the first part is true (since there are two and together) that's the assignment (1, 1, 0) and we can study the only case when the second part is false (since there are two or together) that's the assignment (1, 1, 0) and since this is the same assignment and there's an OR we can see it as A or !A so it is a TAUTOLOGY.

(vi) $(Smoke \implies Fire) \implies ((Smoke \land Heat) \implies Fire)$

!(!Smoke or Fire) or (!(Smoke and Heat) or Fire) ->
(Smoke and !Fire) or (!Smoke or !Heat or Fire) -> (Smoke or !Smoke) and (!Fire or Fire) -> tautology

(vii) $Big \lor Dumb \lor (Big \implies Dumb)$

Big or Dumb or (!Big or Dumb) -> Big or !Big -> tautology

(c) Suppose an agent inhabits a world with two states, S and $\neg S$, and can do exactly one of two actions, a and b. Action a does nothing and action b flips from one state to the other. Let S^t be the proposition that the agent is in state S at time t, and let a^t be the proposition that the agent does action a at time t (similarly for b^t).

(i) Write a successor-state axiom for S^{t+1}.

given a and s, we have that
(a,s)
(0,0)->1
(0,1)->0
(1,0)->0
(1,1)->1
that's equivalent to !(a xor s) that is
!((a or s) and (!a or !s)) -> !(a or s) or !(!a or !s) -> (!a and !s) or (a and s)

(ii) Convert the sentence in the previous part into CNF.

(!a and !s) or (a and s) -> ((!a and !s) or a) and ((!a and !s) or s) ->
((!a or a) and (!s or a)) and ((!a or s) and (!s or s)) ->
(!s or a) and (!a or s)

Q2. First Order Logic

Consider a vocabulary with the following symbols:

- $Occuption(p, o)$: Predicate. Person p has occuption o.
- $Customer(p1, p2)$: Predicate. Person $p1$ is a customer of person $p2$.
- $Boss(p1, p2)$: Predicate. Person $p1$ is a boss of person $p2$.
- $Doctor, Surgeon, Lawyer, Actor$: Constants denoting occupations.
- $Emily, Joe$: Constants denoting people.

Use these symbols to write the following assertions in first-order logic:

(iii) Emily is either a surgeon or a lawyer.

Occupation(Emily, surgeon) or Occupation(Emily, lawyer)

(iv) Joe is an actor, but he also holds another job.

Occupation(Joe, actor) and (exists o != actor such that Occupation(Joe, o))

(v) All surgeons are doctors.

For every p such that occupation(p, surgeon) then occupation(p, doctor)

(vi) Joe does not have a lawyer (i.e., is not a customer of any lawyer).

not customer(Joe, p) for every p occupation(p, lawyer)

(vii) Emily has a boss who is a lawyer.

Exists p occupation (p, lawyer) such that boss(p, Emily)

(viii) There exists a lawyer all of whose customers are doctors.

Exists p occupation(p, lawyer) such that for every c customer(c, p) and occupation(c, doctor)

(ix) Every surgeon has a lawyer.

For every s occupation(s, surgeon), exists l such that customer(s, l) and occupation(l, lawyer)

Q3. [Optional] Local Search

(a) Hill Climbing

(i) Hill-climbing is complete. ☐ True ☑ False

(ii) Hill-climbing is optimal. ☐ True ☑ False

(b) Simulated Annealing

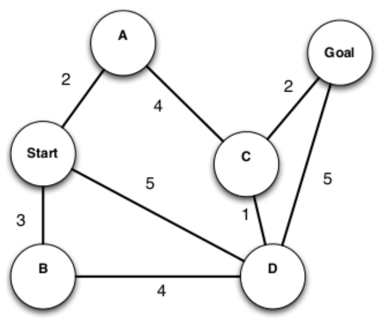
(i) The higher the temperature T is, the more likely the randomly chosen state will be expanded. ☑ True ☐ False

(ii) In one round of simulated annealing, the temperature is 2 and the current state S has energy 1. It has 3 successors: A with energy 2; B with energy 1; C with energy 1-ln 4. If we assume the temperature does not change, What's the probability that these states will be chosen to expand after S eventually?

(iii) On a undirected graph, If T decreases slowly enough, simulated annealing is guaranteed to converge to the optimal state. ☑ True ☐ False

(c) Local Beam Search

The following state graph is being explored with 2-beam graph search. A state's score is its accumulated distance to the start state and lower scores are considered better. Which of the following statements are true?



☑ States A and B will be expanded before C and D.

☐ States A and D will be expanded before B and C.

☐ States B and D will be expanded before A and C.

☐ None of above.

(d) Genetic Algorithm

(i) In genetic algorithm, cross-over combine the genetic information of two parents to generate new offspring.
☑ True ☐ False

(ii) In genetic algorithm, mutation involves a probability that some arbitrary bits in a genetic sequence will be flipped from its original state.
☑ True ☐ False

(e) Gradient Descent

(i) Gradient descent is optimal. ☐ True ☑ False

(ii) For a function $f(x)$ with derivative $f'(x)$, write down the gradient descent update to go from x_t to x_{t+1}. Learning rate is α.

x_t+1 = x_t - alpha*f'(x_t)