CS 188 Spring 2021 | Introduction to Artificial Intelligence | **Final Review Logic**

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 $(\alpha \wedge \beta) \models \gamma$

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

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

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

(i) $Smoke \implies Smoke$

(ii) $Smoke \implies Fire$

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

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

(v) $((Smoke \wedge Heat) \implies Fire)((Smoke \implies Fire) \vee (Heat \implies Fire))$

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

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

(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}.

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

Q2. Disjunctive Normal Form

A sentence is in disjunctive normal form (DNF) if it is the disjunction of conjunctions of literals. For example, the sentence $(A \wedge B \wedge \neg C) \vee (\neg A \wedge C) \vee (B \wedge \neg C)$ is in DNF.

(a) Any propositional logic sentence is logically equivalent to the assertion that some possible world in which it would be true is in fact the case. From this observation, prove that any sentence can be written in DNF.

(b) Construct an algorithm that converts any sentence in propositional logic into DNF. (*Hint*: The algorithm is similar to the algorithm for conversion to CNF.)

(c) Construct a simple algorithm that takes as input a sentence in DNF and returns a satisfying assignment if one exists, or reports that no satisfying assignment exists.

(d) Apply the algorithms in the previous two parts to the following set of sentences:

$$A \implies B$$
$$B \implies C$$
$$C \implies \neg A$$

(e) Since the algorithm in (b) is very similar to the algorithm for conversion to CNF, and since the algorithm in (c) is much simpler than any algorithm for solving a set of sentences in CNF, why is this technique not used in automated reasoning?