Assignment on Propositional Logic: Solutions

Question 1 (Ex.7.5, Russell & Norvig) (3 points)

Give the number of models for each of the following sentences.

(a) $(A \wedge B) \vee (B \wedge C)$

(b) $A \vee B$

(c) $(A \leftrightarrow B) \leftrightarrow C$

Solution

(a) 3, (b) 3, (c) 4

Question 2 (Ex.7.8, Russell & Norvig) (8 points)

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

(a) $Smoke \rightarrow Smoke$

(b) $Smoke \rightarrow Fire$

(c) $(Smoke \rightarrow Fire) \rightarrow (\neg Smoke \rightarrow \neg Fire)$

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

(e) $((Smoke \wedge Heat) \rightarrow Fire) \leftrightarrow ((Smoke \rightarrow Fire) \vee (Heat \rightarrow Fire))$

(f) $(Smoke \rightarrow Fire) \rightarrow ((Smoke \wedge Heat) \rightarrow Fire)$

(g) $Smoke \vee Fire \vee (Smoke \rightarrow Fire)$

(h) $(Smoke \wedge Fire) \vee \neg Fire$

Solution

(a) valid, (b) neither, (c) neither, (d) valid, (e) valid, (f) valid, (g) valid, (h) neither

Question 3 (Ex.7.4, Russell & Norvig) (2 points)

For arbitrary propositional sentences α and β, recall the definition of entailment: $\alpha \models \beta$ if every model of α is a model of β. Prove the following theorem: $\alpha \models \beta$ if and only if $\alpha \wedge \neg\beta$ is unsatisfiable.

Solution

($\Rightarrow$) Assume $\alpha \models \beta$. Now if $\alpha \wedge \neg\beta$ has a model, that model must satisfy α, and falsify β. This contradicts the assumption that every model of α is a model of β. ($\Leftarrow$) Assume $\alpha \wedge \neg\beta$ is unsatisfiable. Then every model of α must falsify $\neg\beta$, or else it would be a model of $\alpha \wedge \neg\beta$, contradicting the assumption. This means that every model of α is a model of β.

Question 4 (Ex.12.11.4, Aho & Ullman) (6 points)

Using the theorem from Question 3, prove each of the following assertions by first converting $\alpha \wedge \neg\beta$ to CNF and then deriving the empty clause by resolution. For better readability you may write a CNF formula as a list of clauses (omitting the $\wedge$).

(a) $(P \rightarrow Q) \wedge (P \rightarrow R) \models P \rightarrow (Q \wedge R)$

(b) $(P \rightarrow (Q \vee R)) \wedge (P \rightarrow (Q \vee \neg R)) \models P \rightarrow Q$

(c) $(P \rightarrow Q) \wedge ((Q \wedge R) \rightarrow S) \models (P \wedge R) \rightarrow S$

Solution (the clauses for each problem are given here and the resolution steps omitted)

(a) $\neg P \vee Q, \neg P \vee R, P, \neg Q \vee \neg R$

(b) $\neg P \vee Q \vee R, \neg P \vee Q \vee \neg R, P, \neg Q$

(c) $\neg P \vee Q, \neg Q \vee \neg R \vee S, P, R, \neg S$

Question 5 (6 points)

Recall the notion of an implication graph described in class, and that each cut through the graph that puts the conflict on one side and all the decisions on the other corresponds to a candidate clause to be learned. Show that each of these candidate clauses, if not already existing, can be derived by resolution from the existing clauses.

Solution

The smallest possible number of nodes on the conflict side of a cut is one, where the candidate clause is the existing clause that has just been falsified giving rise to the conflict. All other cuts can be obtained by "pulling" nodes, one at a time, from the other side to the conflict side, provided each node "pulled" has an arrow going through the cut. We shall show that the pulling of each node corresponds to resolving the current candidate clause with an existing clause, which will prove the result.

Let p be the literal represented by the pulled node, and α the conjunction of the rest of the literals such that $\overline{p} \vee \overline{\alpha}$ is the candidate clause corresponding to the current cut. Let β be the conjunction of literals having arrows going into the pulled node (i.e., that representing literal p). The semantics of the arrows implies that $\overline{\beta} \vee p$ is an existing clause. After literal p gets pulled across the cut, the new candidate clause becomes $\overline{\beta} \vee \overline{\alpha}$, which is precisely the resolvent of $\overline{\beta} \vee p$ and $\overline{p} \vee \overline{\alpha}$.

Question 6 (3 points)

Recall the 1-UIP learning method described in class, where each learned clause contains exactly one literal that was falsified in the latest decision level. Show that this method will never learn a clause that already exists.

Solution

If the learned 1-UIP clause already existed, it would have become a unit clause forcing the instantiation of the UIP literal in a decision level prior to the latest decision level. This would contradict the fact that the UIP literal was instantiated in the latest decision level.

Question 7 (12 points)

Convert the CNF formula $(A \vee B) \wedge (B \vee C)$ to a logically equivalent Boolean circuit in

(a) d-DNNF (deterministic decomposable NNF);

(b) DNNF (decomposable NNF) that does *not* satisfy determinism;

(c) d-NNF (deterministic NNF) that does *not* satisfy decomposability;

(d) sd-DNNF (smooth d-DNNF).

Solution

(a) $B \vee (\neg B \wedge A \wedge C)$

(b) $B \vee (A \wedge C)$

(c) $(A \vee (\neg A \wedge B)) \wedge (B \vee (\neg B \wedge C))$

(d) $(B \wedge (A \vee \neg A) \wedge (C \vee \neg C)) \vee (\neg B \wedge A \wedge C)$