CS 340 Home Work # 5 Solution
CLO 2
Total marks: 80

Q 1. (10 Marks) Consider a vocabulary with only four propositions, A, B, C, and D. How many models *(truth table assignments for which the sentence is true)* are there for the following sentences?

a. $B \vee C$.
b. $\neg A \vee \neg B \vee \neg C \vee \neg D$.
c. $(A \Rightarrow B) \wedge A \wedge \neg B \wedge C \wedge D$.
d. $(A \wedge B) \vee (C \wedge D)$.
e. $B \Rightarrow (A \wedge B)$.

Solution.

These can be computed by counting the rows in a truth table that come out true, but each has some simple property that allows a short-cut:

a. Sentence is false only if B and C are false, which occurs in 4 cases for A and D, leaving 12.

b. Sentence is false only if A, B, C, and D are false, which occurs in 1 case, leaving 15.

c. The last four conjuncts specify a model in which the first conjunct is false, so 0.

d. 4 worlds satisfy $A \wedge B$, 4 satisfy $C \wedge D$, minus 1 for double-counting the model that satisfies both, leaving 7.

e. The sentence is true when B is false (8) and when B is true and A is true (4), so 12 in all.

Q2. (15 marks) Which of the following are correct?

a. $False \models True$.
b. $True \models False$.
c. $(A \wedge B) \models (A \Leftrightarrow B)$.
d. $A \Leftrightarrow B \models A \vee B$.
e. $A \Leftrightarrow B \models \neg A \vee B$.
f. $(A \wedge B) \Rightarrow C \models (A \Rightarrow C) \vee (B \Rightarrow C)$.
g. $(C \vee (\neg A \wedge \neg B)) \equiv ((A \Rightarrow C) \wedge (B \Rightarrow C))$.
h. $(A \vee B) \wedge (\neg C \vee \neg D \vee E) \models (A \vee B)$.
i. $(A \vee B) \wedge (\neg C \vee \neg D \vee E) \models (A \vee B) \wedge (\neg D \vee E)$.
j. $(A \vee B) \wedge \neg(A \Rightarrow B)$ is satisfiable.
k. $(A \Leftrightarrow B) \wedge (\neg A \vee B)$ is satisfiable.
l. $(A \Leftrightarrow B) \Leftrightarrow C$ has the same number of models as $(A \Leftrightarrow B)$ for any fixed set of proposition symbols that includes A, B, C.

Solution.

In all cases, the question can be resolved easily by referring to the definition of entailment.

a. $False \models True$ is true because $False$ has no models and hence entails every sentence AND because $True$ is true in all models and hence is entailed by every sentence.

b. $True \models False$ is false.

c. $(A \wedge B) \models (A \Leftrightarrow B)$ is true because the left-hand side has exactly one model that is one of the two models of the right-hand side.

d. $A \Leftrightarrow B \models A \vee B$ is false because one of the models of $A \Leftrightarrow B$ has both A and B false, which does not satisfy $A \vee B$.

e. $A \Leftrightarrow B \models \neg A \vee B$ is true because the RHS is $A \Rightarrow B$, one of the conjuncts in the definition of $A \Leftrightarrow B$.

f. $(A \wedge B) \Rightarrow C \models (A \Rightarrow C) \vee (B \Rightarrow C)$ is true because the RHS is false only when both disjuncts are false, i.e., when A and B are true and C is false, in which case the LHS is also false. This may seem counterintuitive, and would not hold if $\Rightarrow$ is interpreted as "causes."

g. $(C \vee (\neg A \wedge \neg B)) \equiv ((A \Rightarrow C) \wedge (B \Rightarrow C))$ is true; proof by truth table enumeration, or by application of distributivity (Fig 7.11).

h. $(A \vee B) \wedge (\neg C \vee \neg D \vee E) \models (A \vee B)$ is true; removing a conjunct only allows more models.

i. $(A \vee B) \wedge (\neg C \vee \neg D \vee E) \models (A \vee B) \wedge (\neg D \vee E)$ is false; removing a disjunct allows fewer models.

j. $(A \vee B) \wedge \neg(A \Rightarrow B)$ *is* satisfiable; model has A and $\neg B$.

k. $(A \Leftrightarrow B) \wedge (\neg A \vee B)$ *is* satisfiable; RHS is entailed by LHS so models are those of $A \Leftrightarrow B$.

l. $(A \Leftrightarrow B) \Leftrightarrow C$ does have the same number of models as $(A \Leftrightarrow B)$; half the models of $(A \Leftrightarrow B)$ satisfy $(A \Leftrightarrow B) \Leftrightarrow C$, as do half the non-models, and there are the same numbers of models and non-models.

Q3. (5 Marks) A propositional 2-CNF expression is a conjunction of clauses, each containing exactly 2 literals, e.g.,

$$(A \vee B) \wedge (\neg A \vee C) \wedge (\neg B \vee D) \wedge (\neg C \vee G) \wedge (\neg D \vee G) .$$

Prove using resolution that the above sentence entails G.

Solution.

The negated goal is $\neg G$. Resolve witht he last two clauses to produce $\neg C$ and $\neg D$. Resolve with the second and third clauses to produce $\neg A$ and $\neg B$. Resolve these successively against the first clause to produce the empty clause.

Q4. (10 Marks) Consider the following sentence:

$$[(Food \Rightarrow Party) \vee (Drinks \Rightarrow Party)] \Rightarrow [(Food \wedge Drinks) \Rightarrow Party] .$$

a. Determine, using enumeration, whether this sentence is valid, satisfiable (but not valid), or unsatisfiable.

b. Convert the left-hand and right-hand sides of the main implication into CNF, showing each step, and explain how the results confirm your answer to (a).

c. Prove your answer to (a) using resolution.

Solution.

a. A simple truth table has eight rows, and shows that the sentence is true for all models and hence valid.

b. For the left-hand side we have:

$$\begin{aligned}
&(Food \;\Rightarrow\; Party) \lor (Drinks \;\Rightarrow\; Party)\\
&(\neg Food \lor Party) \lor (\neg Drinks \lor Party)\\
&(\neg Food \lor Party \lor \neg Drinks \lor Party)\\
&(\neg Food \lor \neg Drinks \lor Party)
\end{aligned}$$

and for the right-hand side we have

$$\begin{aligned}
&(Food \land Drinks) \;\Rightarrow\; Party\\
&\neg(Food \land Drinks) \lor Party\\
&(\neg Food \lor \neg Drinks) \lor Party\\
&(\neg Food \lor \neg Drinks \lor Party)
\end{aligned}$$

The two sides are identical in CNF, and hence the original sentence is of the form $P \;\Rightarrow\; P$, which is valid for any P.

c. To prove that a sentence is valid, prove that its negation is unsatisfiable. I.e., negate it, convert to CNF, use resolution to prove a contradiction. We can use the above CNF result for the LHS.

$$\begin{aligned}
&\neg[[(Food \;\Rightarrow\; Party) \lor (Drinks \;\Rightarrow\; Party)] \;\Rightarrow\; [(Food \land Drinks) \;\Rightarrow\; Party]]\\
&[(Food \;\Rightarrow\; Party) \lor (Drinks \;\Rightarrow\; Party)] \land \neg[(Food \land Drinks) \;\Rightarrow\; Party]\\
&(\neg Food \lor \neg Drinks \lor Party) \land Food \land Drinks \land \neg Party
\end{aligned}$$

Each of the three unit clauses resolves in turn against the first clause, leaving an empty clause.

Q5. (10 Marks) Which of the following are valid (necessarily true) sentences?

a. $(\exists x\; x = x) \;\Rightarrow\; (\forall y\;\; \exists z\; y = z)$.

b. $\forall x\;\; P(x) \lor \neg P(x)$.

c. $\forall x\;\; Smart(x) \lor (x = x)$.

Solution.

Validity in first-order logic requires truth in all possible models:

a. $(\exists x\; x = x) \;\Rightarrow\; (\forall y\;\; \exists z\; y = z)$.
Valid. The LHS is valid by itself—in standard FOL, every model has at least one object; hence, the whole sentence is valid iff the RHS is valid. (Otherwise, we can find a model where the LHS is true and the RHS is false.) The RHS is valid because for every value of y in any given model, there is a z—namely, the value of y itself—that is identical to y.

b. $\forall x\;\; P(x) \lor \neg P(x)$.
Valid. For any relation denoted by P, every object x is either in the relation or not in it.

c. $\forall x\;\; Smart(x) \lor (x = x)$.
Valid. In every model, every object satisfies $x = x$, so the disjunction is satisfied regardless of whether x is smart.

Q6. (10 Marks)

Consider a vocabulary with the following symbols:

$Occupation(p, o)$: Predicate. Person p has occupation 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:

a. Emily is either a surgeon or a lawyer.
b. Joe is an actor, but he also holds another job.
c. All surgeons are doctors.
d. Joe does not have a lawyer (i.e., is not a customer of any lawyer).
e. Emily has a boss who is a lawyer.
f. There exists a lawyer all of whose customers are doctors.
g. Every surgeon has a lawyer.

Solution.

a. $O(E, S) \lor O(E, L)$.
b. $O(J, A) \land \exists p \ \ p \neq A \land O(J, p)$.
c. $\forall p \ \ O(p, S) \ \Rightarrow \ O(p, D)$.
d. $\neg \exists p \ \ C(J, p) \land O(p, L)$.
e. $\exists p \ \ B(p, E) \land O(p, L)$.
f. $\exists p \ \ O(p, L) \land \forall q \ \ C(q, p) \ \Rightarrow \ O(q, D)$.
g. $\forall p \ \ O(p, S) \ \Rightarrow \ \exists q \ \ O(q, L) \land C(p, q)$.

Q7. (20 Marks) In each of the following we give an English sentence and a number of candidate logical expressions. For each of the logical expressions, state whether it (1) correctly expresses the English sentence; (2) is syntactically invalid and therefore meaningless; or (3) is syntactically valid but does not express the meaning of the English sentence.

a. Every cat loves its mother or father.
(i) $\forall x \ \ Cat(x) \ \Rightarrow \ Loves(x, Mother(x) \lor Father(x))$.
(ii) $\forall x \ \ \neg Cat(x) \lor Loves(x, Mother(x)) \lor Loves(x, Father(x))$.
(iii) $\forall x \ \ Cat(x) \land (Loves(x, Mother(x)) \lor Loves(x, Father(x)))$.

b. Every dog who loves one of its brothers is happy.
(i) $\forall x \ \ Dog(x) \land (\exists y \ Brother(y, x) \land Loves(x, y)) \ \Rightarrow \ Happy(x)$.

(ii) $\forall x, y \;\; Dog(x) \wedge Brother(y, x) \wedge Loves(x, y) \;\Rightarrow\; Happy(x)$.

(iii) $\forall x \;\; Dog(x) \wedge [\forall y \;\; Brother(y, x) \;\Leftrightarrow\; Loves(x, y)] \;\Rightarrow\; Happy(x)$.

c. No dog bites a child of its owner.

(i) $\forall x \;\; Dog(x) \;\Rightarrow\; \neg Bites(x, Child(Owner(x)))$.

(ii) $\neg\exists x, y \;\; Dog(x) \wedge Child(y, Owner(x)) \wedge Bites(x, y)$.

(iii) $\forall x \;\; Dog(x) \;\Rightarrow\; (\forall y \;\; Child(y, Owner(x)) \;\Rightarrow\; \neg Bites(x, y))$.

(iv) $\neg\exists x \;\; Dog(x) \;\Rightarrow\; (\exists y \;\; Child(y, Owner(x)) \wedge Bites(x, y))$.

d. Everyone's zip code within a state has the same first digit.

(i) $\forall x, s, z_1 \;\; [State(s) \wedge LivesIn(x, s) \wedge Zip(x) = z_1] \;\Rightarrow$
$[\forall y, z_2 \;\; LivesIn(y, s) \wedge Zip(y) = z_2 \;\Rightarrow\; Digit(1, z_1) = Digit(1, z_2)]$.

(ii) $\forall x, s \;\; [State(s) \wedge LivesIn(x, s) \wedge \exists z_1 \;\; Zip(x) = z_1] \;\Rightarrow$
$[\forall y, z_2 \;\; LivesIn(y, s) \wedge Zip(y) = z_2 \wedge Digit(1, z_1) = Digit(1, z_2)]$.

(iii) $\forall x, y, s \;\; State(s) \wedge LivesIn(x, s) \wedge LivesIn(y, s) \;\Rightarrow\; Digit(1, Zip(x) = Zip(y))$.

(iv) $\forall x, y, s \;\; State(s) \wedge LivesIn(x, s) \wedge LivesIn(y, s) \;\Rightarrow$
$Digit(1, Zip(x)) = Digit(1, Zip(y))$.

Solution.

a. Every cat loves its mother or father.

(i) $\forall x \;\; Cat(x) \;\Rightarrow\; Loves(x, Mother(x) \vee Father(x))$.
(2) Syntactically invalid. Cannot have a disjunction inside a term.

(ii) $\forall x \;\; \neg Cat(x) \vee Loves(x, Mother(x)) \vee Loves(x, Father(x))$.
(1) Correct. (Rewrite as implication with disjunctive consequence.)

(iii) $\forall x \;\; Cat(x) \wedge (Loves(x, Mother(x)) \vee Loves(x, Father(x)))$.
(3) Incorrect. Use of $\wedge$ with $\forall$ means that everything is asserted to be a cat.

b. Every dog who loves one of its brothers is happy.

(i) $\forall x \;\; Dog(x) \wedge (\exists y \; Brother(y, x) \wedge Loves(x, y)) \;\Rightarrow\; Happy(x)$.
(1) Correct.

(ii) $\forall x, y \;\; Dog(x) \wedge Brother(y, x) \wedge Loves(x, y) \;\Rightarrow\; Happy(x)$.
(1) Correct. Logically equivalent to (i).

(iii) $\forall x \;\; Dog(x) \wedge [\forall y \;\; Brother(y, x) \;\Leftrightarrow\; Loves(x, y)] \;\Rightarrow\; Happy(x)$.
(3) Incorrect. States that dogs are happy if they love all of, and only, their brothers.

c. No dog bites a child of its owner.

(i) $\forall x \;\; Dog(x) \;\Rightarrow\; \neg Bites(x, Child(Owner(x)))$.
(3) Incorrect. Uses *Child* as a function instead of a relation.

(ii) $\neg\exists x, y \;\; Dog(x) \wedge Child(y, Owner(x)) \wedge Bites(x, y)$.
(1) Correct.

(iii) $\forall x \;\; Dog(x) \;\Rightarrow\; (\forall y \;\; Child(y, Owner(x)) \;\Rightarrow\; \neg Bites(x, y))$.
(1) Correct. Logically equivalent to (ii).

(iv) $\neg\exists x \;\; Dog(x) \;\Rightarrow\; (\exists y \;\; Child(y, Owner(x)) \wedge Bites(x, y))$.
(3) Incorrect. Uses $\Rightarrow$ with $\exists$.

d. Everyone's zip code within a state has the same first digit.

(i) $\forall\, x, s, z_1 \;\; [State(s) \land LivesIn(x, s) \land Zip(x) = z_1] \;\Rightarrow$
$[\forall\, y, z_2 \;\; LivesIn(y, s) \land Zip(y) = z_2 \;\Rightarrow\; Digit(1, z_1) = Digit(1, z_2)]$.
(1) Correct.

(ii) $\forall\, x, s \;\; [State(s) \land LivesIn(x, s) \land \exists\, z_1 \;\; Zip(x) = z_1] \;\Rightarrow$
$[\forall\, y, z_2 \;\; LivesIn(y, s) \land Zip(y) = z_2 \land Digit(1, z_1) = Digit(1, z_2)]$.
(2) Syntactically invalid. Uses z_1 outside scope of its quantifier. Also uses $\land$ as the main connective in the universally quantified RHS.

(iii) $\forall\, x, y, s \;\; State(s) \land LivesIn(x, s) \land LivesIn(y, s) \;\Rightarrow\; Digit(1, Zip(x) = Zip(y))$.
(2) Syntactically invalid. Cannot use equality within a term.

(iv) $\forall\, x, y, s \;\; State(s) \land LivesIn(x, s) \land LivesIn(y, s) \;\Rightarrow$
$Digit(1, Zip(x)) = Digit(1, Zip(y))$.
(1) Correct. Since Zip is a function, there is no need to define additional variables to name the zip codes.