CSE 311: Foundations of Computing I

Equivalences and Predicate Logic 2 Solutions

Equivalences

Prove that each of the following pairs of propositional formulae are equivalent using propositional equivalences.

(a) $(p \to q) \land (q \to p) \qquad (p \land q) \lor (\neg p \land \neg q)$

Solution:

$$
\begin{aligned}
(p \to q) \land (q \to p) &\equiv (\neg p \lor q) \land (q \to p) && \text{[Law of Implication]}\\
&\equiv (\neg p \lor q) \land (\neg q \lor p) && \text{[Law of Implication]}\\
&\equiv ((\neg p \lor q) \land \neg q) \lor ((\neg p \lor q) \land p) && \text{[Distributivity]}\\
&\equiv (\neg q \land (\neg p \lor q)) \lor ((\neg p \lor q) \land p) && \text{[Commutativity]}\\
&\equiv ((\neg q \land \neg p) \lor (\neg q \land q)) \lor ((\neg p \lor q) \land p) && \text{[Distributivity]}\\
&\equiv ((\neg p \land \neg q) \lor (\neg q \land q)) \lor ((\neg p \lor q) \land p) && \text{[Commutativity]}\\
&\equiv ((\neg p \land \neg q) \lor (q \land \neg q)) \lor ((\neg p \lor q) \land p) && \text{[Commutativity]}\\
&\equiv ((\neg p \land \neg q) \lor (q \land \neg q)) \lor (p \land (\neg p \lor q)) && \text{[Commutativity]}\\
&\equiv ((\neg p \land \neg q) \lor (q \land \neg q)) \lor ((p \land \neg p) \lor (p \land q)) && \text{[Distributivity]}\\
&\equiv ((\neg p \land \neg q) \lor F) \lor ((p \land \neg p) \lor (p \land q)) && \text{[Negation]}\\
&\equiv ((\neg p \land \neg q) \lor F) \lor (F \lor (p \land q)) && \text{[Negation]}\\
&\equiv (\neg p \land \neg q) \lor (F \lor (p \land q)) && \text{[Identity]}\\
&\equiv (\neg p \land \neg q) \lor ((p \land q) \lor F) && \text{[Commutativity]}\\
&\equiv (\neg p \land \neg q) \lor (p \land q) && \text{[Identity]}\\
&\equiv (p \land q) \lor (\neg p \land \neg q) && \text{[Commutativity]}
\end{aligned}
$$

(b) $\neg p \to (q \to r) \qquad q \to (p \lor r)$

Solution:

$$
\begin{aligned}
\neg p \to (q \to r) &\equiv \neg\neg p \lor (q \to r) && \text{[Law of Implication]}\\
&\equiv p \lor (q \to r) && \text{[Double Negation]}\\
&\equiv p \lor (\neg q \lor r) && \text{[Law of Implication]}\\
&\equiv (p \lor \neg q) \lor r && \text{[Associativity]}\\
&\equiv (\neg q \lor p) \lor r && \text{[Commutativity]}\\
&\equiv \neg q \lor (p \lor r) && \text{[Associativity]}\\
&\equiv q \to (p \lor r) && \text{[Law of Implication]}
\end{aligned}
$$

Non-equivalence

Prove that the following pairs of propositional formulae are not equivalent by finding inputs they differ on.

(a) $p \rightarrow q$ $\qquad\qquad$ $q \rightarrow p$

Solution:

When $p = \mathsf{T}$ and $q = \mathsf{F}$, then $p \rightarrow q \equiv \mathsf{F}$, but $q \rightarrow p \equiv \mathsf{T}$.

(b) $p \rightarrow (q \wedge r)$ $\qquad\qquad$ $(p \rightarrow q) \wedge r$

Solution:

When $p = \mathsf{F}$ and $r = \mathsf{F}$, then $p \rightarrow (q \wedge r) \equiv \mathsf{T}$, but $(p \rightarrow q) \wedge r \equiv \mathsf{F}$.

Boolean Algebra

For each of the following parts, write the logical expression using boolean algebra operators. Then, simplify it using axioms and theorems of boolean algebra.

(a) $\neg p \vee (\neg q \vee (p \wedge q))$

Solution:

First, we replace $\neg, \vee$, and $\wedge$. This gives us $p' + q' + pq$; note that the parentheses are not necessary in boolean algebra, because the operations are all commutative and associative. We can use DeMorgan's laws to get the slightly simpler $(pq)' + pq$. Then, we can use commutativity to get $pq + (pq)'$ and complementarity to get 1. (Note that this is another way of saying the formula is a tautology.)

(b) $\neg(p \vee (q \wedge p))$

Solution:

First, we replace $\neg, \vee$, and $\wedge$ with their corresponding boolean operators, giving us $(p + (qp))'$. Applying DeMorgan's laws once gives us $p'(qp)'$, and a second time gives us $p'(q' + p')$, which is $p'(p' + q')$ by commutativity. By absorbtion, this is simply p'.

Canonical Forms

Consider the boolean functions $F(A, B, C)$ and $G(A, B, C)$ specified by the following truth table:

A	B	C	$F(A, B, C)$	$G(A, B, C)$
1	1	1	1	0
1	1	0	1	1
1	0	1	0	0
1	0	0	0	0
0	1	1	1	1
0	1	0	1	0
0	0	1	0	1
0	0	0	1	0

(a) Write the DNF and CNF expressions for $F(A, B, C)$.

Solution:

DNF: $ABC + ABC' + A'BC + A'BC' + A'B'C'$
CNF: $(A' + B + C')(A' + B + C)(A + B + C')$

(b) Write the DNF and CNF expressions for $G(A, B, C)$.

Solution:

DNF: $ABC' + A'BC + A'B'C$
CNF: $(A' + B' + C')(A' + B + C')(A' + B + C)(A + B' + C)(A + B + C)$

Translate to Logic

Express each of these English sentences into logical expressions using predicates, quantifiers, and logical connectives.

(a) A cuttlefish is poisonous only if it has spots and does not eat shrimp.

Solution:

Let the domain be cuttlefish. Let HasSpots(x) be "x has spots", EatsShrimp(x) be "x eats shrimp", and IsPoisonous(x) be "x is poisonous".

$$\forall x\ (\text{IsPoisonous}(x) \to (\text{HasSpots}(x) \wedge \neg\text{EatsShrimp}(x)))$$

(b) Every sea creature has either fins or a shell, but not both, unless the sea creature is a cuttlefish.

Solution:

Let the domain be all sea creatures. Let HasFins(x) be "x has fins", HasShell(x) be "x has a shell", Cuttlefish(x) be "x is a cuttlefish".

$$\forall x\ ((\text{HasShell}(x) \oplus (\text{HasFins}(x)) \vee \text{Cuttlefish}(x)))$$

(c) When a shark meets a fish, that shark will eat that fish only if it is not also a shark.

Solution:

Let the domain be all fish (which includes sharks). Let $\text{Shark}(x)$ be "x is a shark", $\text{Meets}(x, y)$ be "x meets y", and $\text{Eats}(x, y)$ be "x eats y".

$$\forall x, y\ ((\text{Shark}(x) \land \text{Meets}(x, y) \land \text{Eats}(x, y) \rightarrow \neg\text{Shark}(y)))$$

Translate to English

Translate these system specifications into English where $F(p)$ is "Printer p is out of service", $B(p)$ is "Printer p is busy", $L(j)$ is "Print job j is lost," and $Q(j)$ is "Print job j is queued". Let the domain be all printers and print jobs.

(a) $\exists p\ (F(p) \land B(p)) \rightarrow \exists j\ L(j)$

Solution:

If at least one printer is busy and out of service, then at least one job is lost.

(b) $(\forall p\ B(p)) \rightarrow (\exists j\ Q(j))$

Solution:

If all printers are busy, then there is a queued job.

(c) $\exists j\ (Q(j) \land L(j)) \rightarrow \exists p\ F(p)$

Solution:

If there is a queued job that is lost, then a printer is out of service.

(d) $(\forall p\ B(p) \land \forall j\ Q(j)) \rightarrow \exists j\ L(j)$

Solution:

If all printers are busy and all jobs are queued, then there is some lost job.

Quantifier Switch

Consider the following pairs of sentences. For each pair, determine if one implies the other (or if they are equivalent).

(a) $\forall x\, \forall y\, P(x,y)$ $\qquad\qquad$ $\forall y\, \forall x\, P(x,y)$

Solution:

These sentences are the same; switching universal quantifiers makes no difference.

(b) $\exists x\, \exists y\, P(x,y)$ $\qquad\qquad$ $\exists y\, \exists x\, P(x,y)$

Solution:

These sentences are the same; switching existential quantifiers makes no difference.

(c) $\forall x\, \exists y\, P(x,y)$ $\qquad\qquad$ $\forall y\, \exists x\, P(x,y)$

Solution:

These are only the same if P is symmetric (e.g. the order of the arguments doesn't matter). If the order of the arguments does matter, then these are different statements. For instance, if $P(x,y)$ is "$x < y$", then the first statement says "for every x, there is a corresponding y such that $x < y$", whereas the second says "for every y, there is a corresponding x such that $x < y$". In other words, in the first statement y is a function of x, and in the second x is a function of y.

(d) $\forall x\, \exists y\, P(x,y)$ $\qquad\qquad$ $\exists x\, \forall y\, P(x,y)$

Solution:

These two statements are usually different.

Positively Different

For $a \in \mathbb{R}$ and $f : \mathbb{R} \to \mathbb{R}$, show that (a) and (b) have different meanings.

(a) $\forall y\ ((y > 0) \to \exists z\ ((z > 0) \land ((|x - a| < z) \to (|f(x) - f(a)| < y)))$

(b) $\exists z\ ((z > 0) \land \forall y\ ((y > 0) \to ((|x - a| < z) \to (|f(x) - f(a)| < y)))$

Solution:

Consider x, y, and z being in the domain of all negative numbers. Then for a), the left-hand side of the first implication evaluates to false ($y > 0$ is false in the domain of all negative numbers). Since $F \to T$ and $F \to F$ both evaluate to true, regardless of what the right-hand side of the first implication evaluates to, the entire expression will be true.

For b), the assertion that there exists a value z such that $z > 0$ is false in the domain of all negative numbers. Since this assertion is connected to the rest of the statement with $\land$, $F \land T$ and $F \land F$ both evaluate to false. Thus, regardless of what the rest of the statement evalutes to, the entire expression will always evaluate to false.

Because we have shown that for the same domain, a) and b) evaluate to different values, it follows that they do not have the same meaning.

TR∀NSL∀TOR

Express each of these sentences using predicates, quantifiers, and logical connectives. Make sure to define a domain for each part.

(a) There are at least two fluffy dogs in every happy house.

Solution:

Let the domain be all houses and dogs. We define the following predicates:

- Let House(x) be "x is a house"
- Let Happy(x) be "x is happy"
- Let Dog(x) be "x is a dog"
- Let Fluffy(x) be "x is fluffy"
- Let LivesIn(x, y) be "x lives in y"

We also define FD(x) as an abbreviation for Fluffy(x) $\land$ Dog(x) to help preserve space.

$$\forall h\ ((\text{House}(h) \land \text{Happy}(h)) \to \exists x\ \exists y\ (\text{FD}(x) \land \text{FD}(y) \land \text{LivesIn}(x, h) \land \text{LivesIn}(y, h) \land \neg\text{Equal}(x, y)))$$

(b) If there is a new book or a cheap book by my favorite author in the bookstore, then I will buy it.

Solution:

Let the domain be all books and authors. We define the following predicates and constants:

- Let Book(x) be "x is a book"
- Let New(x) be "x is new"
- Let Cheap(x) be "x is cheap"
- Let Buy(x) be "I will buy x"

- Let WrittenBy(x, y) be "x is written by y"
- Let FavoriteAuthor be a constant representing my favorite author.

$$\forall b\ ((\text{Book}(b) \land \text{WrittenBy}(b, \text{FavoriteAuthor}) \land (\text{New}(b) \lor \text{Cheap}(b))) \rightarrow \text{Buy}(b))$$

(c) All parks have at least one duck pond with more than one duck.

Solution:

Let the domain be all parks, ducks, and ponds. We define the following predicates:

- Let Park(x) be "x is a park"
- Let Duck(x) be "x is a duck"
- Let Pond(x) be "x is a pond"
- Let Contains(x, y) be "x contains y"

We also define DC(d, x) as an abbreviation for Duck$(x) \land$ Contains(d, x) to help preserve space.

$$\forall p\ (\text{Park}(p) \rightarrow \exists d\ (\text{Pond}(d) \land \text{Contains}(p, d) \land \exists x\ \exists y\ (\text{DC}(d, x) \land \text{DC}(d, y) \land \neg\text{Equal}(x, y))))$$