CSE 311: Foundations of Computing I

Section 2: Equivalences and Predicate Logic Solutions

0. Equivalences

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

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

Solution:

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

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

Solution:

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

1. Non-equivalence

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

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

Solution:

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

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

Solution:

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

2. 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'.

3. 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: $(A' + B' + C')(A' + B + C')(A' + B + C)(A + B' + C)(A + B + C)$

4. Translate to Logic

Express each of these system specifications using predicate, quantifiers, and logical connectives.

(a) Every user has access to an electronic mailbox.

Solution:

Let the domain be users and mailboxes. Let User(x) be "x is a user", let Mailbox(y) be "y is a mailbox", and let Access(x,y) be "x has access to y".

$$\forall x\ (\text{User}(x) \to \exists y\ (\text{Mailbox}(y) \land \text{Access}(x,y)))$$

(b) The system mailbox can be accessed by everyone in the group if the file system is locked.

Solution:

Let the domain be people in the group and all mailboxes. Let Access(x,y) be "x has access to y". Let FileSystemLocked be the proposition "the file system is locked." Let SystemMailbox be the constant that is the system mailbox.

$$\text{FileSystemLocked} \to \forall x\ \text{Access}(x, \text{SystemMailbox})$$

(c) The firewall is in a diagnostic state only if the proxy server is in a diagnostic state.

Solution:

Let the domain be all applications. Let Firewall(x) be "x is the firewall", and let ProxyServer(x) be "x is the proxy server." Let Diagnostic(x) be "x is in a diagnostic state".

$$\forall x\ \forall y\ ((\text{Firewall}(x) \wedge \text{Diagnostic}(x)) \rightarrow (\text{ProxyServer}(y) \rightarrow \text{Diagnostic}(y)))$$

(d) At least one router is functioning normally if the throughput is between 100kbps and 500 kbps and the proxy server is not in diagnostic mode.

Solution:

Let the domain be all applications and routers. Let Router(x) be "x is a router", and let ProxyServer(x) be "x is the proxy server." Let Diagnostic(x) be "x is in a diagnostic state". Let ThroughputNormal be "the throughput is between 100kbps and 500 kbps". Let Functioning(y) be "y is functioning normally".

$$\forall x\ ((\text{ThroughputNormal} \wedge (\text{ProxyServer}(x) \wedge \neg\text{Diagnostic}(x))) \rightarrow \exists y\ (\text{Router}(y) \wedge \text{Functioning}(y)))$$

5. 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) $\exists p\ (F(p) \wedge B(p)) \rightarrow \exists j\ L(j)$

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) $\exists j\ (Q(j) \wedge L(j)) \rightarrow \exists p\ F(p)$

Solution:

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

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

Solution:

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

6. 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.

7. 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 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.

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

(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.

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