CSE 311: Foundations of Computing I

Section 2: Equivalences and Predicate Logic Solutions

1. Equivalences

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

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

Solution:

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

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

Solution:

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

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".

$$\forall x\ (\text{User}(x) \rightarrow (\exists y\ (\text{Mailbox}(y) \wedge \text{Access}(x, y))))$$

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

Solution:

Solution1: Let the domain be people in the group. Let CanAccessSM(x) be "x has access to the system mailbox". Let FileSystemLocked be the proposition (predicate that is just a constant function) "the file system is locked."

$$\text{FileSystemLocked} \rightarrow \forall x\ \text{CanAccessSM}(x).$$

Solution2: Let the domain be people and mailboxes and use Access(x, y) as defined in the solution to part (a), and then also add InGroup(x) for "x is in the group", and let SystemMailBox be the name for the system mailbox. Then the translation becomes

$$\text{FileSystemLocked} \rightarrow \forall x\ (\text{InGroup}(x) \rightarrow \text{Access}(x, \text{SystemMailBox})).$$

(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".

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

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 all print jobs.

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

Solution:

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

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

Solution:

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

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

Solution:

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

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

Solution:

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