CSC238:Discrete Mathematics for Computer Science – Spring 2003

Solutions for Assignment 3.

---

1. State whether or not each of the following assertions is true, and justify your answer (in particular, you can use a truth table):

   (a) $(x \to y) \wedge (x \to z)$ is logically equivalent to $x \to (y \wedge z)$.

   (b) $(y \to x) \wedge (z \to x)$ is logically equivalent to $(y \wedge z) \to x$.

   Answers:

   (a) The assertion is true:

$$\begin{array}{llll} (x \to y) \wedge (x \to z) & \text{LEQV} & (\neg x \vee y) \wedge (\neg x \vee z) & \text{by } \to \text{ law, twice,} \\ & \text{LEQV} & \neg x \vee (y \wedge z) & \text{by distributive law,} \\ & \text{LEQV} & x \to (y \wedge z) & \text{by } \to \text{ law.} \end{array}$$

   (b) The assertion "$(y \to x) \wedge (z \to x)$ is logically equivalent to $(y \wedge z) \to x$" is false. To prove this, suppose τ is a truth assignment for which $\tau(x)$, $\tau(y)$, $\tau(z)$ are 0, 0, 1, respectively (where 0 denotes false and 1 denotes true). Then $z \to x$ is not satisfied since z is true and x is false. Therefore $(y \to x) \wedge (z \to x)$ must also be false in this truth assignment since it is the conjunction with a false statement. Similarly, the antecedent in the second expression, namely $y \wedge z$, is not satisfied for this same truth assignment. Thus the second expression $(y \wedge z) \to x$ must itself be satisfied. Therefore we see that $\tau^*((y \to x) \wedge (z \to x)) = 0$ while $\tau^*((y \wedge z) \to x) = 1$. Hence the two expressions have different truth values for this single truth assignment τ, and therefore they cannot be logically equivalent.

---

2. Use only the logical equivalences in Section 5.6 of the lecture notes (and, in particular, **do not use truth tables**) to prove:

   (a) Show $(x \leftrightarrow \neg y) \to z$ is logically equivalent to $(x \wedge y) \vee (\neg x \wedge \neg y) \vee z$.

Answer. First we consider the formula

$$\begin{array}{lll}
\neg(x \leftrightarrow \neg y) & \text{LEQV} \ \neg((x \wedge \neg y) \vee (\neg x \wedge \neg\neg y)) & \text{by } \leftrightarrow \text{ law,} \\
& \text{LEQV} \ \neg((x \wedge \neg y) \vee (\neg x \wedge y)) & \text{by } \neg\neg \text{ law,} \\
& \text{LEQV} \ \neg(x \wedge \neg y) \wedge \neg(\neg x \wedge y) & \text{by De Morgan's law,} \\
& \text{LEQV} \ (\neg x \vee \neg\neg y) \wedge (\neg\neg x \vee \neg y) & \text{by De Morgan's law,} \\
& \text{LEQV} \ (\neg x \vee y) \wedge (x \vee \neg y) & \text{by } \neg\neg \text{ law,} \\
& \text{LEQV} \ (x \vee \neg y) \wedge (\neg x \vee y) & \text{by commutative law,} \\
& \text{LEQV} \ ((x \vee \neg y) \wedge \neg x) \vee ((x \vee \neg y) \wedge y) & \text{by distributive law, with} \\
& & P = (x \vee \neg y),\ Q = \neg x,\ R = y, \\
& \text{LEQV} \ (\neg x \wedge (x \vee \neg y)) \vee (y \wedge (x \vee \neg y)) & \text{by commutative law, twice,} \\
& \text{LEQV} \ ((\neg x \wedge x) \vee (\neg x \wedge \neg y)) \vee & \\
& \qquad ((y \wedge x) \vee (y \wedge \neg y)) & \text{by distributive law, twice,} \\
& \text{LEQV} \ (\neg x \wedge \neg y) \vee (y \wedge x) & \text{by idempotent law, twice,} \\
& \text{LEQV} \ (x \wedge y) \vee (\neg x \wedge \neg y) & \text{by commutative laws.}
\end{array}$$

Therefore

$$\begin{array}{lll}
(x \leftrightarrow \neg y) \to z & \text{LEQV} \ \neg(x \leftrightarrow \neg y) \vee z & \text{by } \to \text{ law,} \\
& \text{LEQV} \ ((x \wedge y) \vee (\neg x \wedge \neg y)) \vee z & \text{from derivation above,} \\
& \text{LEQV} \ (x \wedge y) \vee (\neg x \wedge \neg y) \vee z & \text{from associative law.}
\end{array}$$

(b) Show $(x \leftrightarrow \neg y) \to \neg(x \to y)$ is logically equivalent to $y \to x$.

Answer. Note that we can substitute a formula P for variable z in the derivation in part (a), to obtain a proof that

$$(x \leftrightarrow \neg y) \to P \ \text{ LEQV } \ (x \wedge y) \vee (\neg x \wedge \neg y) \vee P$$

for any propositional formula P. Let P be defined to be $\neg(x \to y)$. Notice that

$$\begin{array}{lll}
\neg(x \to y) & \text{LEQV} \ \neg(\neg x \vee y) & \text{by } \to \text{ law,} \\
& \text{LEQV} \ \neg\neg x \wedge \neg y & \text{by De Morgan's law,} \\
& \text{LEQV} \ x \wedge \neg y & \text{by } \neg\neg \text{ law.}
\end{array}$$

Therefore P is logically equivalent to P', where P' is defined to be $x \wedge \neg y$. It follows from Thm 5.10 that

$$\begin{array}{ll}
(x \leftrightarrow \neg y) \to \neg(x \to y) & \text{LEQV} \ (x \leftrightarrow \neg y) \to P \text{ by defn of } P, \\
& \text{LEQV} \ (x \wedge y) \vee (\neg x \wedge \neg y) \vee P \text{ by argument above,} \\
& \text{LEQV} \ (x \wedge y) \vee (\neg x \wedge \neg y) \vee P' \text{ since } P \text{ LEQV } P', \\
& \text{LEQV} \ (x \wedge y) \vee (\neg x \wedge \neg y) \vee (x \wedge \neg y) \text{ by defn } P', \\
& \text{LEQV} \ (x \wedge y) \vee (\neg y \wedge (x \vee \neg x)) \text{ by comm. and dist. laws,} \\
& \text{LEQV} \ (x \wedge y) \vee \neg y \ \text{ by idempotent law,} \\
& \text{LEQV} \ (\neg y \vee x) \wedge (\neg y \vee y) \text{ by comm. and dist. laws,} \\
& \text{LEQV} \ \neg y \vee x \ \text{ by comm. and idempotent laws,} \\
& \text{LEQV} \ y \to x \ \text{ by } \to \text{ law.}
\end{array}$$

(c) Show $(x \wedge \neg y) \to \neg z$ is logically equivalent to $(x \wedge z) \to y$.

Answer:

$$\begin{array}{rlll} (x \wedge \neg y) \to \neg z & \text{LEQV} & \neg(x \wedge \neg y) \vee \neg z & \text{by } \to \text{ law,} \\ & \text{LEQV} & (\neg x \vee \neg\neg y) \vee \neg z & \text{by De Morgan's law,} \\ & \text{LEQV} & \neg x \vee y \vee \neg z & \text{by assoc. and } \neg\neg \text{ laws,} \\ & \text{LEQV} & (\neg x \vee \neg z) \vee y & \text{by comm. and assoc. laws,} \\ & \text{LEQV} & \neg(x \wedge z) \vee y & \text{by De Morgan's law,} \\ & \text{LEQV} & (x \wedge z) \to y & \text{by } \to \text{ law.} \end{array}$$

---

3. Suppose P and Q are two unstatisfiable propositional formulas, R is a tautology, and S is satisfiable. For each of the statements below, decide whether or not it is true, and prove your answer.

   (a) $P \leftrightarrow Q$ is a tautology.

   (b) P logically implies R.

   (c) R logically implies P.

   (d) $P \leftrightarrow R$ is unsatisfiable.

   (e) $S \to P$ is satisfiable.

   Answers: In each of the following parts, let τ be any truth assignment. Since P and Q are unstatifiable, it follows that $\tau^*(P) = \tau^*(Q) = 0$ (i.e. P and Q are not satisfied). Also, since R is a tautology, it follows that $\tau^*(R) = 1$ (i.e. R is satisfied).

   (a) It is true that $P \leftrightarrow Q$ is a tautology.

   Proof. Let τ be any truth assignment. By the definition of $\leftrightarrow$ we know $\tau^*(P \leftrightarrow Q) = 1$ when $\tau^*(P) = \tau^*(Q)$. In the preamble above, we showed that $\tau^*(P) = \tau^*(Q) = 0$. So therefore $\tau^*(P \leftrightarrow Q) = 1$. Equivalently, this means that $P \leftrightarrow Q$ is satisfied for this truth assignment. Since this is true for an arbitrary truth assignment τ, we conclude $P \leftrightarrow Q$ is a tautology.

   (b) It is true that P logically implies R.

   Proof. Let τ be any truth assignment. From the preamble above, recall that $\tau^*(P) = 0$. That is, no truth assigment satisfies P. Therefore it is trivially the case that every truth assignment that satisfies P must also satisfy R. Therefore P logically implies R.

   (c) It is false that R logically implies P.

   Proof. Let τ be any truth assignment. Then from the preamble above, $\tau^*(R) = 1$ and $\tau^*(P) = 0$. That is, τ is a truth assignment which satisfies R but not P. This proves that R does not logically imply P.

(d) It is true that $P \leftrightarrow R$ is unsatisfiable.

Proof. Let τ be an arbitrary truth assignment. From the preamble above we have $\tau^*(P) = 0 \neq 1 = \tau^*(R)$. Therefore $\tau^*(P \leftrightarrow R) = 0$. Since this is true for any truth assignment, we conclude that $P \leftrightarrow R$ is unsatisfiable.

(e) It is not necessarily true that $S \to P$ is satisfiable.

Proof. Suppose S is both satisfiable and a tautology. Let τ be an arbitrary truth assignment. From the preamble above we have $\tau^*(P) = 0$. Also, by the definition of a tautology, $\tau^*(S) = 1$. Therefore $\tau^*(S \to P) = 0$. Since this is true for any truth assignment, we conclude that $S \to P$ is not satisfiable in the special case for which S is a tautology.

Notice that in order for $S \to P$ to be satisfiable, we require $\neg S$ to be satisfiable and this does not follow from the satisfiability of S.

---

4. Write two propositional formulas, one in DNF the other in CNF, that are logically equivalent to $((x \to y) \lor (\neg x \land z)) \leftrightarrow (y \lor z)$. Justify your answers.

Answer. The truth table is:

x	y	z	$((x \to y)$	$\lor$	$(\neg x \land z))$	$\leftrightarrow$	$(y \lor z)$
0	0	0	1	1	0	0	0
0	0	1	1	1	1	1	1
0	1	0	1	1	0	1	1
0	1	1	1	1	1	1	1
1	0	0	0	0	0	1	0
1	0	1	0	0	0	0	1
1	1	0	1	1	0	1	1
1	1	1	1	1	0	1	1

The truth table shows that the formula is true for every row except the following two: the row for x, y, z all false; and the row for x and z true, but y false. A CNF formula can be obtained from the maxterms corresponding to these two rows, namely

$$(x \lor y \lor z) \land (\neg x \lor y \lor \neg z).$$

To obtain a DNF formula a disjunction of minterms can be formed, with one minterm term for each of the six rows of the truth table for which the expression is true. Alternatively, it can be noted that 4 of these 6 rows are associated with y being true (and all combinations of values for the other propositional variables). Hence the disjunction of the four corresponding minterms is equivalent to just the expression y. The remaining two rows for which the expression is true are given by: 1) x and y both false but z true; and 2) x true but y and z false. The minterms for these latter two rows are $\neg x \wedge \neg y \wedge z$ and $x \wedge \neg y \wedge \neg z$, respectively. Combined with the minterm y for the other four rows, we have:

$$y \vee (\neg x \wedge \neg y \wedge z) \vee (x \wedge \neg y \wedge \neg z).$$

This can be further simplified to (proof omitted)

$$y \vee (\neg x \wedge z) \vee (x \wedge \neg z).$$

---

5. Prove that $\{\downarrow\}$ (i.e. nor, as described in the lecture notes) is a complete set of connectives.

Proof. Define $\mathcal{D}$ to be the set of propositional formulas which only use the connectives $\{\neg, \vee\}$. That is, $\mathcal{D}$ is the smallest set such that:

Base Case: Any propositional variable is in $\mathcal{D}$.
Induction Step: If F and G are formulas in $\mathcal{D}$, then so are $\neg F$ and $(F \vee G)$.

By Theorem 5.24 $\{\neg, \vee\}$ is a complete set of connectives. Therefore any propositional formula is logically equivalent to a formula in $\mathcal{D}$.

Similarly, let $\mathcal{N}$ be the set of propositional formulas on the same set of variables, but which only use the connective $\downarrow$. To be precise, define $\mathcal{N}$ to be the smallest set such that

Base Case: Any propositional variable is in $\mathcal{N}$.
Induction Step: If F' and G' are formulas in $\mathcal{N}$, then so is $(F' \downarrow G')$.

We will prove that $\{\downarrow\}$ is complete by using structural induction to show that we can rewrite any formula in $\mathcal{D}$ as a logically equivalent formula in $\mathcal{N}$.

Base Case: Suppose the formula $F \in \mathcal{D}$ is simply a propositional variable. Then by the base case for $\mathcal{N}$, we find $F' = F \in \mathcal{N}$.

Induction Hypothesis: Suppose F and G are two formulas in $\mathcal{D}$, and suppose F' and G' are two formulas in $\mathcal{N}$ such that F and F' are logically equivalent, and G and G' are logically equivalent.

Induction Step: From the induction step in the definition of $\mathcal{D}$ above, we see that we need to show that there are two formulas in $\mathcal{N}$ that are logically equivalent to $\neg F$ and $(F \vee G)$.

Proof of Induction Step: Let us first consider the formula $\neg F$. Notice that $\neg F$ is logically equivalent to $(F \downarrow F)$ (i.e. when F is true, $(F \downarrow F)$ must be false, and vice versa). By the Induction Hypothesis we also have $F' \in \mathcal{N}$ is logically equivalent to F. Using Theorem 5.10 we therefore we find that $\neg F$ is logically equivalent to $(F' \downarrow F')$. But, by the definition of $\mathcal{N}$ and the fact that $F' \in \mathcal{N}$, it must be the case that $(F' \downarrow F') \in \mathcal{N}$. This proves that there is a formula in $\mathcal{N}$, namely $(F' \downarrow F')$, which is logically equivalent to $\neg F$.

The remaining case is the formula $(F \vee G)$. By the Induction Hypothesis and Thm. 5.10 this is logically equivalent to $(F' \vee G')$, where each of F' and G' are in $\mathcal{N}$. Using a truth table, say, it is easy to verify that $(P \vee Q)$ is logically equivalent to the formula $((P \downarrow Q) \downarrow (P \downarrow Q))$, for any propositional formulas P and Q. Consider H' to be the formula obtained using F' for P and G' for Q, that is, H' denotes $((F' \downarrow G') \downarrow (F' \downarrow G'))$. Given this definition, and the logical equivalence exhibited above, we find that $(F' \vee G')$ is logically equivalent to H'. Finally, since we previously showed that $(F \vee G)$ and $(F' \vee G')$ were logically equivalent, we conclude that H' is logically equivalent to $(F \vee G)$.

Next we show that $H' \in \mathcal{N}$. Indeed, by the Induction Hypothesis $F', G' \in \mathcal{N}$. Therefore, by the inductive step in the definition of $\mathcal{N}$, we know that $(F' \downarrow G')$ must be in $\mathcal{N}$. Applying the same inductive step again, but this time to the formula $(F' \downarrow G')$, we find that $H' \in \mathcal{N}$, as desired.

Therefore we have constructed $H' \in \mathcal{N}$ which is logically equivalent to $(F \vee G)$. This proves the inductive step for this second (and last) case.

It now follows by structural induction that, for any formula $F \in \mathcal{D}$ there exists a logically equivalent formula $F' \in \mathcal{N}$. Since $\mathcal{D}$ is known to be complete, it follows that $\mathcal{N}$ must also be complete. That is $\{\downarrow\}$ is a complete set of connectives.

---

6. Is $\{\oplus, \wedge, \vee\}$ a complete set of connectives? Here $\oplus$ denotes the "exclusive or" connective described in the lecture notes. Prove your answer.

   No it is not. A proof is as follows:

   Let $\mathcal{C}$ be the set of propositional formulas that only use the connectives $\{\oplus, \wedge, \vee\}$. To be precise, $\mathcal{C}$ is defined to be the smallest set such that

   > **Base Case:** Any propositional variable is in $\mathcal{C}$.
   > **Induction Step:** If F and G are formulas in $\mathcal{C}$, then so are $(F \oplus G)$, $(F \wedge G)$, and $(F \vee G)$.

   We will use the following lemma in our proof:

   Lemma. Let τ be the truth assignment for which each propositional variable is set to be false. Suppose P is any formula in $\mathcal{C}$. Then $\tau^*(P) = 0$. That is, P is not satisfied for this truth assignment τ.

From this Lemma it is easy to prove that $\{\oplus, \wedge, \vee\}$ cannot be a complete set of connectives. Notice that the propositional formula $\neg x$ is satisfied by the truth assignment τ which sets all propositional variables to be false (including x). Therefore $\tau^*(\neg x) = 1$. Since the Lemma ensures that $\tau^*(P) = 0$ for any $P \in \mathcal{C}$, we conclude that $\neg x$ cannot be logically equivalent to any formula P in $\mathcal{C}$. Therefore the connectives $\{\oplus, \wedge, \vee\}$ cannot be complete.

We are left with proving the Lemma.

Proof of Lemma. We use structural induction on P.

Base Case: Suppose P is a propositional variable. Then by the assumption that τ sets every propositional variable to be false, we have $\tau^*(P) = 0$, as desired.

Induction Hypothesis: Suppose $P, Q \in \mathcal{C}$ and $\tau^*(P) = \tau^*(Q) = 0$.

Induction Step: We need to show $\tau^*(P') = 0$ where P' is any one of the formulas $(P \oplus Q)$, $(P \wedge Q)$, and $(P \vee Q)$. Therefore there are three cases.

In each case the inductive hypothesis ensures that $\tau^*(P) = \tau^*(Q) = 0$, i.e. P and Q are not satisfied.

Case 1: P' is $(P \oplus Q)$. By the definition of $\oplus$, $(P \oplus Q)$ is not satisfied when neither P nor Q are satisfied. Therefore $\tau^*((P \oplus Q)) = 0$, as desired.

Case 2: P' is $(P \wedge Q)$. By the definition of $\wedge$, $(P \wedge Q)$ is not satisfied when neither P nor Q are satisfied. Therefore $\tau^*((P \wedge Q)) = 0$, as desired.

Case 3: P' is $(P \vee Q)$. By the definition of $\vee$, $(P \vee Q)$ is not satisfied when neither P nor Q are satisfied. Therefore $\tau^*((P \vee Q)) = 0$, as desired.

Therefore, in all three cases, $\tau^*(P') = 0$. The proof of the lemma now follows by structural induction.