Mathematics for Computer Science
Exercise session 3, 15 September 2021

Silvio Capobianco

Last update: 23 August 2022

Problems from Section 3.1

Problem 3.3.

When the mathematician says to his student, "If a function is not continuous, then it is not differentiable,", then letting D stand for "differentiable" and C for "continuous", the only proper translation of the mathematician's statement would be

$$\textbf{not}(C) \textbf{ implies } \textbf{not}(D) \,,$$

or equivalently,

$$D \textbf{ implies } C \,.$$

But when a mother says to her son, "If you don't do your homework, then you can't watch TV", then letting T stand for "can watch TV" and H for "do your homework", a reasonable translation of the mother's statement would be

$$\textbf{not}(H) \textbf{ iff } \textbf{not}(T) \,,$$

or equivalently,

$$H \textbf{ iff } T \,.$$

Explain why it is reasonable to translate these two IF-THEN statements in different ways into propositional formulas.

Problem 3.5.

Sloppy Sam is trying to prove a certain proposition P. He defines two related propositions Q and R, and then proceeds to prove three implications:

$$P \text{ implies } Q, \quad Q \text{ implies } R, \quad R \text{ implies } P.$$

He then reasons as follows:

> If Q is true, then since I proved Q **implies** R, I can conclude that R is true. Now, since I proved R **implies** P, I can conclude that P is true. Similarly, if R is true, then P is true and so Q is true. Likewise, if P is true, then so are Q and R. So any way you look at it, all three of P, Q and R are true.

(a) Exhibit truth tables for

$$(P \text{ implies } Q) \text{ and } (Q \text{ implies } R) \text{ and } (R \text{ implies } P) \qquad (1)$$

and for

$$P \text{ and } Q \text{ and } R. \qquad (2)$$

Use these tables to find a truth assignment for P, Q, R so that (1) is **T** and (2) is **F**.

(b) You show these truth tables to Sloppy Sam and he says "OK, I'm wrong that P, Q and R all have to be true, but I still don't see the mistake in my reasoning. Can you help me understand my mistake?" How would you explain to Sammy where the flaw lies in his reasoning?

Exercises for Section 3.3

Problem 3.11.

Indicate whether each of the following propositional formulas is valid (V), satisfiable but not valid (S), or not satisfiable (N). For the satisfiable ones,

indicate a satisfying truth assignment.

$$
\begin{array}{rll}
M & \textbf{implies} & Q \\
M & \textbf{implies} & (\overline{P} \textbf{ or } \overline{Q}) \\
M & \textbf{implies} & (M \textbf{ and } (P \textbf{ implies } M)) \\
(P \textbf{ or } Q) & \textbf{implies} & Q \\
(P \textbf{ or } Q) & \textbf{implies} & (\overline{P} \textbf{ and } \overline{Q}) \\
(P \textbf{ or } Q) & \textbf{implies} & (M \textbf{ and } (P \textbf{ implies } M)) \\
(P \textbf{ xor } Q) & \textbf{implies} & Q \\
(P \textbf{ xor } Q) & \textbf{implies} & (\overline{P} \textbf{ or } \overline{Q}) \\
(P \textbf{ xor } Q) & \textbf{implies} & (M \textbf{ and } (P \textbf{ implies } M)) \\
\end{array}
$$

Problem 3.17.

This problem examines whether the following specifications are *satisfiable*:

1. If the file system is not locked, then

   (a) new messages will be queued.

   (b) new messages will be sent to the messages buffer.

   (c) the system is functioning normally,

   and conversely, if the system is functioning normally, then the file system is not locked.

2. If new messages are not queued, then they will be sent to the messages buffer.

3. New messages will not be sent to the message buffer.

(a) Begin by translating the five specifications into propositional formulas using four propositional variables:

   $L ::=$ file system locked,

   $Q ::=$ new messages are queued,

   $B ::=$ new messages are sent to the message buffer,

   $N ::=$ system functioning normally.

(b) Demonstrate that this set of specifications is satisfiable by describing a single truth assignment for the variables L, Q, B, N and verifying that under this assignment, all the specifications are true.

(c) Argue that the assignment determined in part (b) is the only one that does the job.

Problems for Section 3.4

Problem 3.18(a)-(b).

A half dozen different operators may appear in propositional formulas, but just **and** , **or** , and **not** are enough to do the job. That is because each of the operators is equivalent to a simple formula using only these three operators. For example, A **implies** B is equivalent to **not**(A) **or** B. So all occurences of **implies** in a formula can be replaced using just **not** and **or** .

(a) Write formulas using only **and** , **or** , **not**$()$ that are equivalent to each of A **iff** B and A **xor** B. Conclude that every propositional formula is equivalent to an **and** - **or** -**not** formula.

(b) Explain why you don't even need **and** .

Problems for Section 3.6

Problem 3.26.

For each of the following propositions:

1. $\forall x . \exists y . 2x - y = 0$

2. $\forall x . \exists y . x - 2y = 0$

3. $\forall x . (x < 10 \text{ \textbf{implies} } (\forall y . (y < x \text{ \textbf{implies} } y < 9)))$

4. $\forall x . \exists y . (y > x \land \exists z . y + z = 100)$

determine which propositions are true when the variables range over:

(a) the nonnegative integers,

(b) the integers,

(c) the real numbers.

Problem 3.30.

Find a counter-model showing the following is not valid.

$$\exists x . P(x) \textbf{ implies } \forall x . P(x)$$

(Just define your counter-model. You do not need to verify that it is correct.)

Problem 3.33 (modified).

(a) Verify that the propositional formula

$$(P \textbf{ implies } Q) \textbf{ or } (Q \textbf{ implies } P) \tag{3}$$

is valid.

(b) The valid formula of part (a) leads to sound proof method: to prove that an implication is true, just prove that its converse is false.[1]

But wait a minute! The implication

> If an integer is prime, then it is negative

is completely false. So we could conclude that its converse

> If an integer is negative, then it is prime

should be true, but in fact the converse is also completely false.
So something has gone wrong here. Explain what.

---

[1]This problem was stimulated by the discussion of the fallacy in J. Beam, A Powerful Method of Non-Proof, The College Mathematics Journal 48(1), 52–54.

Solutions

Problem 3.3.

When the mathematician talks to his student, there is no hidden assumption. A function can be discontinuous; or it can be continuous but not differentiable; or it can be differentiable, hence continuous. Implication goes only one direction.

But where the mother talks to the son, there *is* a hidden assumption! Both the mother and the son know that the son will watch TV after he has done his homework. Implication goes both ways.

Problem 3.5.

(a) When we construct the truth tables for (1) and (2), we obtain:

P	Q	R	P **implies** Q	Q **implies** R	R **implies** P	(1)	(2)
T	T	T	T	T	T	T	T
T	T	F	T	F	T	F	F
T	F	T	F	T	T	F	F
T	F	F	F	T	T	F	F
F	T	T	T	T	F	F	F
F	T	F	T	F	T	F	F
F	F	T	T	T	F	F	F
F	F	F	T	T	T	T	F

If P, Q and R are all **F**, then (1) is **T** and (2) is **F**.

(b) Sam is silently assuming that *some* of P, Q and R are true. But why should it be so? All he has proved is that they are equivalent: either they all all true, or all false. To check which is the case, he must find a proof or disproof of any of the three, but at least one of them, and the proof itself must not depend on the others, but only on other things which he *knows*, not just assumes, to be true.

Problem 3.11.

Let's check one by one:

S. M **implies** Q is **T** as soon as M is **F**, but is **F** if M is **T** and Q is **F**.

S. M **implies** $(\overline{P}$ **or** $\overline{Q})$ is **T** as soon as M is **F**, but is **F** if M, P and Q are all **T**.

V. M **implies** $(M$ **and** $(P$ **implies** $M))$ is valid. If M is **F**, then M **implies** $(M$ **and** $(P$ **implies** $M))$ is **T**; if M is **T**, then so is P **implies** M, and so is M **and** $(P$ **implies** $M)$.

S. $(P$ **or** $Q)$ **implies** Q is **T** if Q is **T**, but **F** if P is **T** and Q is **F**.

S. $(P$ **or** $Q)$ **implies** $(\overline{P}$ **or** $\overline{Q})$ is **T** if P and Q are both **F**, but **F** if they are both **T**.

S. $(P$ **or** $Q)$ **implies** $(M$ **and** $(P$ **implies** $M))$ is **T** if P and Q are both **F**, but **F** if they are both **T** and M is **F**.

S. $(P$ **xor** $Q)$ **implies** Q is **T** if P and Q are both **T**, but **F** if P is **T** and Q is **F**.

V. $(P$ **or** $Q)$ **implies** $(\overline{P}$ **or** $\overline{Q})$ is valid. If P and Q are both **T** or both **F**, then the implication from **F** is **T**; if one is **T** and the other is **F**, then one of $\overline{P}$ and $\overline{Q}$ is **T**, and the implication to **T** is **T**.

S. $(P$ **xor** $Q)$ **implies** $(M$ **and** $(P$ **implies** $M))$ is **T** if P and Q are both **F**, but **F** if P is **T** and Q and M are both **F**.

Problem 3.17.

(a) Let us rewrite the three specifications as three Boolean formulas α, β and γ:

1. $\alpha ::= (\mathbf{not}(L)$ **implies** $(Q$ **and** B **and** $N))$ **and** $(N$ **implies** $\mathbf{not}(L))$.

2. $\beta ::= \mathbf{not}(Q)$ **implies** B.

3. $\gamma ::= \mathbf{not}(B)$.

(b) We must find a truth assignment to L, Q, B and N that makes each of α, β and γ take value **T**. We immediately observe that $\gamma = \mathbf{T}$ if and only if $B = \mathbf{F}$. In this case, for β to be **T** it must be $\mathbf{not}(Q) = \mathbf{F}$, hence $Q = \mathbf{T}$.

Now, to have $\alpha = \mathbf{T}$, we must have both

$$\mathbf{not}(L) \textbf{ implies } (Q \textbf{ and } B \textbf{ and } N) = \mathbf{T}$$

and

$$N \textbf{ implies } \mathbf{not}(L) = \mathbf{T}.$$

But the conjunction in the right-hand side of the implication in the first formula is $\mathbf{F}$, because so is B: we must then have $\mathbf{not}(L) = \mathbf{F}$, that is, $L = \mathbf{T}$. Then the second formula can only be $\mathbf{T}$ if $N = \mathbf{F}$.

We then have that the specification is verified if, and only if:

1. the system is locked,

2. new messages are queued,

3. new messages are not sent to the message buffer, and

4. the system does not function normally.

(c) Note that the text of the exercise as reported in the book is imprecise. With that formulation, α becomes:

$$\mathbf{not}(L) \textbf{ implies } (Q \textbf{ and } B \textbf{ and } (N \textbf{ and } (N \textbf{ implies } \mathbf{not}(L))))$$

But this formula *is not* equivalent to

$$(\mathbf{not}(L) \textbf{ implies } (Q \textbf{ and } B \textbf{ and } N)) \textbf{ and } (N \textbf{ implies } \mathbf{not}(L)),$$

because the former is satisfied with $L = Q = \mathbf{T}$ and $B = \mathbf{F}$ regardless of the value of N.

The assignment from part (b) is the only one that does the job because we have *constructed* it starting from the only hypothesis that the formula be satisfied, and at each point in the construction we only had one possible choice to make..

Problem 3.18(a)-(b).

(a) A **iff** B is equivalent to:

$$(A \textbf{ and } B) \textbf{ or } (\mathbf{not}(A) \textbf{ and } \mathbf{not}(B)),$$

while A **xor** B is equivalent to:

$$(A \text{ and } \textbf{not}(B)) \text{ or } (\textbf{not}(A) \text{ and } B),$$

or to:

$$(A \text{ or } B) \text{ and } (\textbf{not}(A) \text{ or } \textbf{not}(B)).$$

Observe how we arrived from the formula for **iff** to the one for **xor** by swapping the roles of **and** and **or** : this is not a case, as A **xor** B is equivalent to **not**(A **iff** B).

(b) That we don't even need **and** follows from de Morgan's law:

$$\textbf{not}(A \text{ and } B) \longleftrightarrow \textbf{not}(A) \text{ or } \textbf{not}(B),$$

which, together with the double negation rule, gives:

$$A \text{ and } B \longleftrightarrow \textbf{not}(\textbf{not}(A) \text{ or } \textbf{not}(B)).$$

Problem 3.26.

(a) If the variables are nonnegative integers:

1. is true, because however given x, we can choose $y = 2x$.

2. is false, because for $x = 1$ the difference $1 - 2y$ is odd for every integer y, and cannot be zero.

3. is true, because if m and n are any two integers, then $m < n$ if and only if $m \leq n - 1$.

4. is false, because if we choose $x = 100$, $y > x$, and z any nonnegative integer, then $y + z > 100$.

(b) If the variables are integers:

1. is true, because however given x, we can choose $y = 2x$.

2. is false, because for $x = 1$ the difference $1 - 2y$ is odd for every integer y, and cannot be zero.

3. is true, because if m and n are any two integers, then $m < n$ if and only if $m \leq n - 1$.

4. is true, because however we choose x and $y > x$, we can always set $z = 100 - y$, which is integer if y is.

(c) If the variables are real numbers:

1. is true, because however given x, we can choose $y = 2x$.

2. is true, because however given x, we can choose $y = x/2$.

3. is false, because for $x = 9.75$ we can take $y = 9.5$, which is smaller than x but larger than 9.

4. is true, because however we choose x and $y > x$, we can always set $z = 100 - y$.

Problem 3.30.

Recall that a *counter-model* of a predicate formula is a setting where the formula is false. In our case, a possible counter-model is:

- the environment is the arithmetics of nonnegative integers,

- the variables range over the nonnegative integers, and

- $P(x) ::= x = 0$.

Recall that quantifiers bind stronger than propositional connectives: therefore, $\exists x . P(x)$ **implies** $\forall x . P(x)$ means $(\exists x . P(x))$ **implies** $(\forall x . P(x))$. Within this counter-model, our predicate means:

> If there is a nonnegative integer equal to zero,
> then every nonnegative integer is equal to zero.

This is clearly false, because there *is* a nonnegative integer equal to zero (the premise is true) but not *every* nonnegative integer is equal to zero (the conclusion is false).

Problem 3.33 (modified).

(a). This is easily done via truth table:

P	Q	$(P \textbf{ implies } Q)$	**or**	$(Q \textbf{ implies } P)$
T	**T**	**T**	**T**	**T**
T	**F**	**F**	**T**	**T**
F	**T**	**T**	**T**	**F**
F	**F**	**T**	**T**	**T**

Alternatively, we can reason by cases, according to P being true or false:

$P = \textbf{T}$. Then Q **implies** P is true, and so is (3).

$P = \textbf{F}$. Then P **implies** Q is true, and so is (3).

(b). The formula which we proved valid is a *propositional* formula: it is true whenever P and Q have a *definite* truth value. This allows us to conclude, for example, that

$$\forall x \,.\, ((P(x) \textbf{ implies } Q(x)) \textbf{ or } (Q(x) \textbf{ implies } P(x))) \qquad (4)$$

is valid: as soon as the value of x is chosen, so are the truth values of $P(x)$ and $Q(x)$, and whatever those are, the proposition in parentheses is true.

However,

<p align="center">If an integer is prime, then it is negative</p>

is not a propositional formula; it is a *predicate* formula, which, in the context of real-valued functions defined on an open interval, corresponds to:

$$\forall x \in \mathbb{Z} \,.\, (P(x) \textbf{ implies } Q(x))$$

where $P(x)$ and $Q(x)$ are interpreted as "x is prime" and "x is negative", respectively.

Now, when we start from the falsity of

<p align="center">If an integer is prime, then it is negative</p>

to "prove"

> If an integer is negative, then it is prime

we are *not* applying (4), but a different formula:

$$(\forall x \,.\, (P(x) \text{ implies } Q(x))) \text{ or } (\forall x \,.\, (Q(x) \text{ implies } P(x))) \quad (5)$$

But the formula (5) *is not* valid in predicate logic: and it cannot be, because we have a counter-model right in front of our eyes!