Mathematics for Computer Science
Exercise session 3, 14 September 2022

Silvio Capobianco

Last update: 20 September 2023

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.

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 3.15.

The formula

$$\begin{gathered}
\textbf{not}(\overline{A} \textbf{ implies } B) \textbf{ and } A \textbf{ and } C \\
\textbf{implies} \\
D \textbf{ and } E \textbf{ and } F \textbf{ and } G \textbf{ and } H \textbf{ and } I \textbf{ and } J \textbf{ and } K \textbf{ and } L \textbf{ and } M
\end{gathered}$$

turns out to be valid.

1. Explain why verifying the validity of this formula *by truth table* would be very hard for one person to do with pencil and paper (no computers).

2. Verify that the formula is valid, reasoning by cases according to the truth value of A.

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:

$$L ::= \text{file system locked},$$
$$Q ::= \text{new messages are queued},$$
$$B ::= \text{new messages are sent to the message buffer},$$
$$N ::= \text{system functioning normally}.$$

(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 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 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** .

(c) Explain how to get by with the single operator **nand** where A **nand** B is equivalent by definition to **not**(A **and** B).

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. $\forall x \,.\, (x < 10 \textbf{ implies } (\forall y \,.\, (y < x \textbf{ implies } y < 9)))$
4. $\forall x \,.\, \exists y \,.\, (y > x \wedge \exists z \,.\, y + z = 100)$

determine which propositions are true when the variables range over:

(a) the nonnegative integers,

(b) the integers,

(c) the real numbers.

Problems 3.29, 3.30, and 3.31.

For each of the following formulas, find a counter-model:

1. $(\forall x \,.\, \exists y \,.\, P(x, y)) \textbf{ implies } \forall z \,.\, P(z, z)$
2. $\exists x \,.\, P(x) \textbf{ implies } \forall x \,.\, P(x)$
3. $(\exists x \,.\, P(x) \textbf{ and } \exists x \,.\, Q(x)) \textbf{ implies } \exists x \,.\, (P(x) \textbf{ and } Q(x))$

Hint: you can always use the arithmetics of natural numbers as a domain, and the set of natural numbers as type for the variables; you only need to find suitable interpretations for the predicates.

Problem 3.33 (modified).

(a) Verify that the propositional formula

$$(P \textbf{ implies } Q) \textbf{ or } (Q \textbf{ implies } P) \tag{1}$$

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.

This page intentionally left blank.

This page too.

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 way.

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 still goes one way, but this time, there are *two* implications.

Problem 3.11.

One by one, in the order given:

S. M **implies** Q is true as soon as M is false, but is false if M is true and Q is false.

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

V. M **implies** (M **and** (P **implies** M)) is valid. If M is false, then M **implies** (M **and** (P **implies** M)) is true; if M is true, then so is P **implies** M, and so is M **and** (P **implies** M).

S. (P **or** Q) **implies** Q is true if Q is true, but false if P is true and Q is false.

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

S. (P **or** Q) **implies** (M **and** (P **implies** M)) is true if P and Q are both false, but false if they are both true and M is false.

S. (P **xor** Q) **implies** Q is true if P and Q are both true, but false if P is true and Q is false.

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

S. (P **xor** Q) **implies** (M **and** (P **implies** M)) is true if P and Q are both false, but false if P is true and Q and M are both false.

Problem 3.15.

1. The formula depends on thirteen propositional variables. A truth table for it would have $2^{13} = 8192$ rows.

2. As the formula is an implication, it is sufficient to prove that the premise

$$\textbf{not}(\overline{A}\ \textbf{implies}\ B)\ \textbf{and}\ A\ \textbf{and}\ C$$

is always false. We do so by reasoning by cases on the truth value of A.

- Case 1: A is true. Then $\overline{A}$ **implies** B is true: correspondingly, **not**($\overline{A}$ **implies** B) is false, and so is **not**($\overline{A}$ **implies** B) **and** A **and** C.
- Case 2: A is false. Then **not**($\overline{A}$ **implies** B) **and** A **and** C is clearly false.

Problem 3.17.

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

(a) $\alpha ::= (\textbf{not}(L)\ \textbf{implies}\ (Q\ \textbf{and}\ B\ \textbf{and}\ N))\ \textbf{and}\ (N\ \textbf{implies}\ \textbf{not}(L))$.

(b) $\beta ::= \textbf{not}(Q)\ \textbf{implies}\ B$.

(c) $\gamma ::= \textbf{not}(B)$.

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

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

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

and

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

But the conjunction in the right-hand side of the implication in the first formula is false, because so is B: we must then have $\textbf{not}(L) = \mathbf{F}$, 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:

   (a) the system is locked,
   (b) new messages are queued,
   (c) new messages are not sent to the message buffer, and
   (d) 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:

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

But this formula *is not* equivalent to

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

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) First, we observe that A **iff** B is true if and only if A and B are either both true, or both false. The first case corresponds to A **and** B being true; the second case corresponds to **not**(A) **and not**(B) being true. Then A **iff** B is equivalent to:

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

Next, we observe that A **iff** B is true if and only if either A is true and B is false, or A is false and B is true. The first case corresponds to A **and not**(B) being true; the second case corresponds to **not**(A) **and** B being true. Then A **xor** B is equivalent to:

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

Alternatively, we could have observed that A **xor** B is equivalent to **not**(A **iff** B).

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

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

which, together with the double negation rule, gives:

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

(c) Because of the previous points, it is sufficient to show that **not**() and **or** can be reconstructed by only using **nand** . One part is easy:

$$\textbf{not}(A) \longleftrightarrow A \textbf{ nand } A\,.$$

For the second one, we use de Morgan's law together with double negation:

$$\begin{aligned} A \textbf{ or } B &\longleftrightarrow \textbf{not}(\textbf{not}(A) \textbf{ and } \textbf{not}(B)) \\ &\longleftrightarrow \textbf{not}(A) \textbf{ nand } \textbf{not}(B) \\ &\longleftrightarrow (A \textbf{ nand } A) \textbf{ nand } (B \textbf{ nand } B)\,. \end{aligned}$$

As a note, since A **and** **T** is equivalent to A, we could also have obtained **not**(A) as A **nand** **T**.

Problem 3.26.

(a) If the variables are nonnegative integers:

(a) is true, because however given x, we can choose $y = 2x$.

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

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

(d) 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:

(a) is true, because however given x, we can choose $y = 2x$.

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

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

(d) 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:

(a) is true, because however given x, we can choose $y = 2x$.

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

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

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

Problems 3.29, 3.30, and 3.31.

1. Interpret $P(x, y)$ as "$x < y$". Then the formula means "if for every natural number there exists a larger natural number, then every natural number is smaller than itself", which is false.

2. Interpret $P(x)$ as "$x = 2$". Then the formula means "if there is a natural number equal to 2, then all natural numbers are equal to 2", which is false.

3. Interpret $P(x)$ as "$x > 17$" and $Q(x)$ as "$x < 17$". Then the formula means "if there exists a natural number larger than 17 and there exists a natural number smaller than 17, then there exists a natural number that is larger er and smaller than 17 at the same time", which is false.

Problem 3.33 (modified).

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

P	Q	(P **implies** Q)	**or**	(Q **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:

$P = \mathbf{T}$. Then Q **implies** P is true, and so is (1).

$P = \mathbf{F}$. Then P **implies** Q is true, and so is (1).

(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

$$\forall x \,.\, ((P(x) \textbf{ implies } Q(x)) \textbf{ or } (Q(x) \textbf{ implies } P(x))) \tag{2}$$

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,

If an integer is prime, then it is negative

is not a propositional formula; it is a *predicate* formula, which, in the context of integer arithmetics, 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

If an integer is prime, then it is negative

to "prove"

If an integer is negative, then it is prime

we are *not* applying (2), but a different formula:

$$(\forall x \,.\, (P(x) \textbf{ implies } Q(x))) \textbf{ or } (\forall x \,.\, (Q(x) \textbf{ implies } P(x))) \tag{3}$$

But the formula (3) *is not* valid in predicate logic: and it cannot be, because we have a counter-model right in front of our eyes!