

Solutions to In-Class Problems Week 1, Fri.

Problem 1. Albert & Ronitt announce that they plan a surprise 6.042 quiz next week. Their students wonder if the quiz could be next Friday. The students realize that it obviously cannot, because if it hadn't been given before Friday, everyone would know that there was only Friday left on which to give it, so it wouldn't be a surprise any more.

So the students ask whether Albert & Ronitt could give the surprise quiz Thursday? They observe that if the quiz wasn't given *before* Thursday, it would have to be given *on* the Thursday, since they already know it can't be given on Friday. But having figured that out, it wouldn't be a surprise if the quiz was on Thursday either. Similarly, the students reason that the quiz can't be on Wednesday, Tuesday, or Monday. Namely, it's impossible for Albert & Ronitt to give a surprise quiz next week. All the students now relax having concluded that Albert & Ronitt must have been bluffing.

And since no one expects the quiz, that's why, when Albert & Ronitt give it on Tuesday next week, it really is a surprise!

What do you think is wrong with the students' reasoning?

Solution. The basic problem is that "surprise" is not a mathematical concept, nor is there any generally accepted way to give it a mathematical definition. The "proof" above assumes some plausible axioms about surprise, without defining it. The paradox is that these axioms are inconsistent. But that's no surprise :-), since—mathematically speaking—we don't know what we're talking about.

Mathematicians and philosophers have had a lot more to say about what might be wrong with the students' reasoning, (see Chow, Timothy Y. *The surprise examination or unexpected hanging paradox*, American Math. Monthly (January 1998), pp.41–51.) ■

Problem 2. Identify the antecedents and conclusions of each of the following deductions and

translate them into propositional logic notation using logical operators:

$$\begin{aligned}\wedge &::= \text{ AND,} \\ \vee &::= \text{ OR,} \\ \neg &::= \text{ NOT,} \\ \longrightarrow &::= \text{ IMPLIES,} \\ \longleftrightarrow &::= \text{ IFF (if and only if)}\end{aligned}$$

This may require that you “pin down” a statement that could be interpreted in more than one way. Identify which of the deductions are sound ones.

(a) Jane and Pete won’t both win the math prize. Pete will win either the math prize or the chemistry prize. Jane will win the math prize. Thus, Pete will win the chemistry prize.

Solution. The deduction is:

$$\frac{\neg(J \wedge M), \quad M \vee C, \quad J}{C}$$

where

$$\begin{aligned}J &::= \text{ “Jane will win the math prize.”} \\ M &::= \text{ “Pete will win the math prize.”} \\ C &::= \text{ “Pete will win the chemistry prize.”}\end{aligned}$$

This deduction is sound. ■

(b) The main course will be beef or fish. The vegetable will be peas or corn. We will not have both fish as a main course and corn as a vegetable. Therefore, we will not have both beef as a main course and peas as a vegetable.

Solution. The deduction is:

$$\frac{B \vee F, \quad C \vee P, \quad \neg(F \wedge C)}{\neg(B \wedge P)}$$

where

$$\begin{aligned}B &::= \text{ “The main course will be beef.”} \\ F &::= \text{ “The main course will be fish.”} \\ C &::= \text{ “The vegetable will be corn.”} \\ P &::= \text{ “The vegetable will be peas.”}\end{aligned}$$

This deduction is not sound. For example, $B \wedge \neg F \wedge C \wedge P$ is consistent with the antecedents but not with the conclusion. Note that as formalized, there need not be only one main course and only one vegetable; it is possible, for example, for the vegetable to be both corn and peas, as in the scenario given.

If we wished to exclude the possibility of multiple courses we could have used *exclusive-or* instead of *inclusive-or*. So our antecedent about the main course would then read $B \oplus F$ or, equivalently, $(B \vee F) \wedge \neg(B \wedge F)$. The antecedent about the vegetable could be changed similarly. The deduction is still unsound in this formalization. ■

(c) Either John or Bill is telling the truth. Either Sam or Bill is lying. Thus, either John is telling the truth or Sam is lying.

Solution. We interpret “John is lying,” to be the negation of “John is telling the truth.” Similarly for the corresponding propositions involving Bill and Sam. The deduction is:

$$\frac{J \vee B, \quad \neg S \vee \neg B}{J \vee \neg S}$$

where

$J ::=$ “John is telling the truth.”

$B ::=$ “Bill is telling the truth.”

$S ::=$ “Sam is telling the truth.”

This deduction is sound. It is an example of a common “cancellation” or *cut* rule that lets us get rid of the proposition B in the conclusion. ■

(d) Either sales will go up and the boss will be happy, or expenses will go up and the boss won't be happy. Therefore, sales and expenses will not both go up.

Solution. The deduction is:

$$\frac{(S \wedge H) \vee (E \wedge \neg H)}{\neg(S \wedge E)}$$

where

$S ::=$ “Sales will go up.”

$H ::=$ “The boss will be happy.”

$E ::=$ “Expenses will go up.”

This deduction is not sound. For example, $S \wedge E \wedge H$ is consistent with the antecedent but not with the conclusion. ■

Problem 3. Boolean logic comes up in digital circuit design using the convention that **T** corresponds to 1 and **F** to 0. For example, suppose we want to describe a circuit with $n + 1$ inputs $a_n, a_{n-1}, \dots, a_1, a_0$ which are the $n + 1$ bits of the binary representation of an integer, k , between 0 and $2^{n+1} - 1$. We want outputs $o_{n+1}, o_n, \dots, o_1, o_0$ to be the bits of $k + b$ where b is a single bit.

For example, for $n = 1$, the formulas

$$o_0 ::= a_0 \oplus b$$

$$c_1 ::= a_0 \wedge b$$

the carry into column 1

$$o_1 ::= c_1 \oplus a_1$$

$$c_2 ::= c_1 \wedge a_1$$

the carry into column 2

$$o_2 ::= c_2$$

do the job. Here \oplus is the “mod 2 sum” operator: $a \oplus b$ is 1 iff $a + b$ is even.

(a) Generalize the example above for any $n \geq 0$. That is, give simple formulas for o_i and c_i for $0 \leq i \leq n + 1$.

Solution. Define

$$\begin{aligned} o_0 &::= a_0 \oplus b, \\ c_{i+1} &::= a_i \wedge c_i && \text{for } 0 \leq i \leq n, \\ o_{i+1} &::= c_{i+1} \oplus a_{i+1} && \text{for } 0 \leq i < n, \\ o_{n+1} &::= c_{n+1}. \end{aligned}$$

■

(b) Write similar definitions for the $n+1$ bits of the sum of two binary numbers $a_n, a_{n-1}, \dots, a_1, a_0$ and $b_n, b_{n-1}, \dots, b_1, b_0$.

Solution. Define

$$\begin{aligned} o_0 &::= a_0 \oplus b_0, \\ c_{i+1} &::= (a_i \wedge b_i) \vee (a_i \wedge c_i) \vee (b_i \wedge c_i) && \text{for } 0 \leq i \leq n, \\ o_{i+1} &::= c_{i+1} \oplus a_{i+1} \oplus b_{i+1} && \text{for } 0 \leq i < n, \\ o_{n+1} &::= c_{n+1}. \end{aligned}$$

■

(c) How many Boolean operations does your system use to calculate the sum?

Solution. The scheme above uses $3(n+1)$ AND'S, $2n+1$ MOD-2-SUMS and $2(n+1)$ OR's for a total of $7n+5$ operations. ■