

Diabologic: Black Tie

by Frank Dolinar

For computer geeks and engineers, it starts out as a joke and travels a path that addresses the nature of humor. For philosophers it looks into the nature of truth. It goes like this...

She asks, "Do you want to have Italian food or Chinese food for dinner?"

He answers, "Yes."

She responds, "You always do that!" – frustrated because she didn't get an expected answer.

But she did get an answer and a perfectly valid one at that.

You may disagree, but in this case (a case of formal logic) your opinion isn't significant.

Why? Because if the choice of either Italian or Chinese (or both) is acceptable, then as a logical equation, e.g. if ((I) or (C)), the equation is logically true. And his answer must be "Yes". Only if neither is acceptable, i.e. both evaluate to be logically false, can his answer be "No". Indeed, in such a case his answer not only can be "No", it must be "No".

This goes back to those "formal logic" courses in college that were required courses for some academic programs. Most people disliked them. I enjoyed them... but I had a secret weapon. (More about that in a forthcoming column.)

At the beginning of the 21st century, our society continues to lean ever more heavily on science and technology to get us through the day. Many processes are controlled by real-time computer programs that never stop running. That may not be absolutely true. There can be situations that cause some forms of shutdown, but in general such programs are designed to keep on running and running...

In today's world, the difference between "yes" (logical TRUE) or "no" (logical FALSE) isn't just an answer, it's a command and control decision. When such evaluations are made, they can lead to situations where money changes hands, industries suffer setbacks (think of BP's broken oil rig in the Gulf of Mexico), or lives hang in the balance.

But let's go back to those computer programs for a moment. Years ago, I encountered a situation at work where a recently modified program consistently generated the wrong answer. All of the code seemed okay, at first glance. So, three of us kept looking at the offending code until... ah ha! A lovely, perfectly legitimate command, happened to contain a logic error that looked like this:

IF (NOT (A AND B)) ... then perform some function... otherwise, don't bother.

Knowing something about this program's intent, however, I suspected a problem.

The construct "Not A and B" can be evaluated two ways. The one above, or like this:

IF ((NOT A) AND B) ... which, depending on the values of A and B, may or may not achieve the same TRUE or FALSE value as the first method.

Regardless, they are separate and distinct ways of looking at the statement, and in this case, the second way was the original intent, but the programmer had misplaced a couple of parentheses.

These are statements in Boolean Logic, named for George Boole, a 19th century English mathematician and philosopher, regarded as a founder of the field of computer science. Boolean logic deals with the truth or falsity of logical statements, the evaluation of which was codified by Boole.

There are five basic evaluation mechanisms: **NOT; OR; NOR; AND; NAND**

- NOT** given a single value, NOT reverses it – IF A is TRUE, NOT A is FALSE
- OR** given two values, OR returns FALSE only if both values are FALSE, otherwise it returns TRUE
- NOR** given two values, NOR returns TRUE only if both values are FALSE, otherwise it returns FALSE, thus exactly reversing any value that would have been returned by an OR
- AND** given two values, AND returns TRUE only if both values are TRUE, otherwise it returns FALSE
- NAND** given two values, NAND returns FALSE only if both values are TRUE, otherwise it returns TRUE, thus exactly reversing any value that would have been returned by an AND

What good is this stuff? Well it's not arithmetic, but it's the fundamental quanta of Set Theory. Set Theory is a foundation of how databases work and an integral part of the Structured Query Language (SQL) of all modern relational database systems.

And databases? Without them, querying Google for information would be impossible.

So, you see, his answer (at the beginning of this article) was not just valid, it was absolutely correct.

Logically.