

# Re: Troolean operators

---

*Source:* <http://sci.tech-archive.net/Archive/sci.physics/2006-03/msg00994.html>

---

- *From:* [rick\\_sobie@xxxxxxxxxxx](mailto:rick_sobie@xxxxxxxxxxx)
  - *Date:* 11 Mar 2006 11:42:53 -0800
- 

rick\_sobie@xxxxxxxxxxx wrote:

"Well when you get out there, in space, be sure that your flaps are not on too steep an angle, you don't want friction to burn them when they are turned towards the sun"

So, OK, there is no atmosphere in space, flaps need air pressure to work, yet would the sun cause them a problem, and would that problem surface, when you are back on earth, trying to land?

If the person was a known liar who told you that, you would be worried because number one, it sounds serious, yet has components which may not make sense.

So it may be a true statement, it may be a false statement, and it may be pure nonsense.

Lets take it a bit further.

So you say, well either what he said was true, or what he said was false.

Granted, so what is the result?

Undefined.

Maybe parts of what he said were true and parts were false, but the whole thing sounds like bs, and you have to make a decision.

So you see, if you stop there, you might crash. So using troolean logic,

## Re: Troolean operators

you would expect the unexpected, and in this case get a second opinion and if everyone is just making stuff up?

So you still need to assume, that it may be true, it may be false, and if something happens that is not what you might expect then you have a plan c.

And that is where a whole lot of problems occur. Right there in plan b.

Because you assume a thing is either true or false, so you have a plan b.

So why stop there? Why not have a plan d, just to be sure?  
By the time you hit plan c, you no longer have any basis for logical deductive reasoning, so to go to plan d, is just a more robust plan c.

case in point, the first attempts to land a probe on the moon.  
Ranger 3, misses the moon, goes into orbit around the sun.  
Ranger 4, piles into the moon at about mach 5.  
Ranger 5, misses the moon by 450 miles.

So OK, what we see here is that ranger 5 doesn't look much like a plan c, it looks like yet a repeat of plan a.

Why was that? Well they assumed the gravity on the moon was according to their physics a certain value, and well it wasn't.

Out of the blue, here is a situation that defied belief.  
But the result was an unexpected occurrence, not accounted for that the gravity on the moon would differ from what everyone would expect at that time.

So they stopped there, Russia was launching a set of probes called Luna at the same time. They too had been unsuccessful. After Ranger 5, Russia launches Luna 7, it bores into the moon full tilt.  
~poof~ (that's the little dust ball that comes up on impact as it disintegrates)

but they kept at it, by now convinced it wasn't just technical glitches and their formulii must be wrong, and so trial and error, and voila, Luna 8 crashes, Luna 9 lands.  
And then America is successful as well, after everyone finds out what the values actualy were.

All along they had been dealing with formulii, that contained a lie.  
And the result was unpredictable accordingly.

It is not sufficient to go on values from a book say as you go along, you need to be able to read those values as input, as you go along,

## Re: Troolean operators

and calibrate your equations based on known facts.

So if Windows, instead of boolean operators, used troolean operators, in every boolean instance, throughout the code, you would need to provide for three possibilities.

This was implemented by the programmers as they went along anyways, but not right there at the place where it needs to be.

Every question, every boolean question, needs a what if.

So rather than say, if this then that, what we now have is a thing called try.

A catch block that goes around your equation, and it says try this... and if there is a problem then it catches it.

Systems are much more reliable and crash far less as a result of that. You don't even see, the myriad of times, the program you are using catches itself from screwing up. It doesn't have to report every error.

It just catches it, and carries on, you push a button, and you go, Hmm... nothing happened, and so you try something else.

.