

Mathematics deals with absolute truth, right? Errr, well, absolute truth is an incredibly loaded term. Properly unpacking it requires a healthy amount of philosophy and mathematics. Wait. Don't go away. It's fun. You'll be grappling with paradoxical statements like 'this sentence is false' in a productive and well-defined way. You'll find statements that are neither true nor false, yet are critical to our modern understanding of mathematics.

1 Philosophical Underpinnings

1.1 won't get fooled again

I'll tip my hat to the new constitution
Take a bow for the new revolution
Smile and grin at the change all around
Pick up my guitar and play
Just like yesterday
Then I'll get on my knees and pray
We don't get fooled again
Don't get fooled again
No, no!

The Who

Rhetoric is an ancient tradition of disguising opinions as sound logic. Through millions of years of evolution, we carbon sack computers have been tuned to take shortcuts. Taking shortcuts in thought is necessary: no one would teach an infant to count by formally proving $2 + 2 = 4$. But with these shortcuts come logical fallacies, passing off emotions as fact, perception as reality.

We could start with assertions and then work toward a conclusion by requiring each new claim we make must logically follow from what we have already established, but English is too ambiguous and irregular. Consider the sentence

Every mouse fears some cat.

Does this mean there is one terribly frightening cat that every single mouse fears? Or does it mean for each mouse there is a corresponding scary cat?

We need to move toward logical deductions governed by rules expressed in an appropriate notation (not English). These rules should be straight-forward enough that they can be checked by a computer, but expressive enough that we can work out all mathematics in terms of these logical deductions.

We look to answer a nebulous philosophical question, 'is mathematics absolutely true?' by building a mathematical model that can handle mathematics, and then asking our model what it can tell us about truth in mathematics. Such an approach discards much nuance of the philosophical question; we're punting. This punting is necessary, because the question 'is mathematics absolutely true' is rife with uncertainty. What is mathematics? What is absolute truth? Does absolute truth exist? How can a human judge truth if 'to err is human'? One can object *ad nauseum*.

There's also a similar problem. Since we cannot get something from nothing, we'll need to use logic in order to set up our mathematical model of logic. This circular reasoning can't be

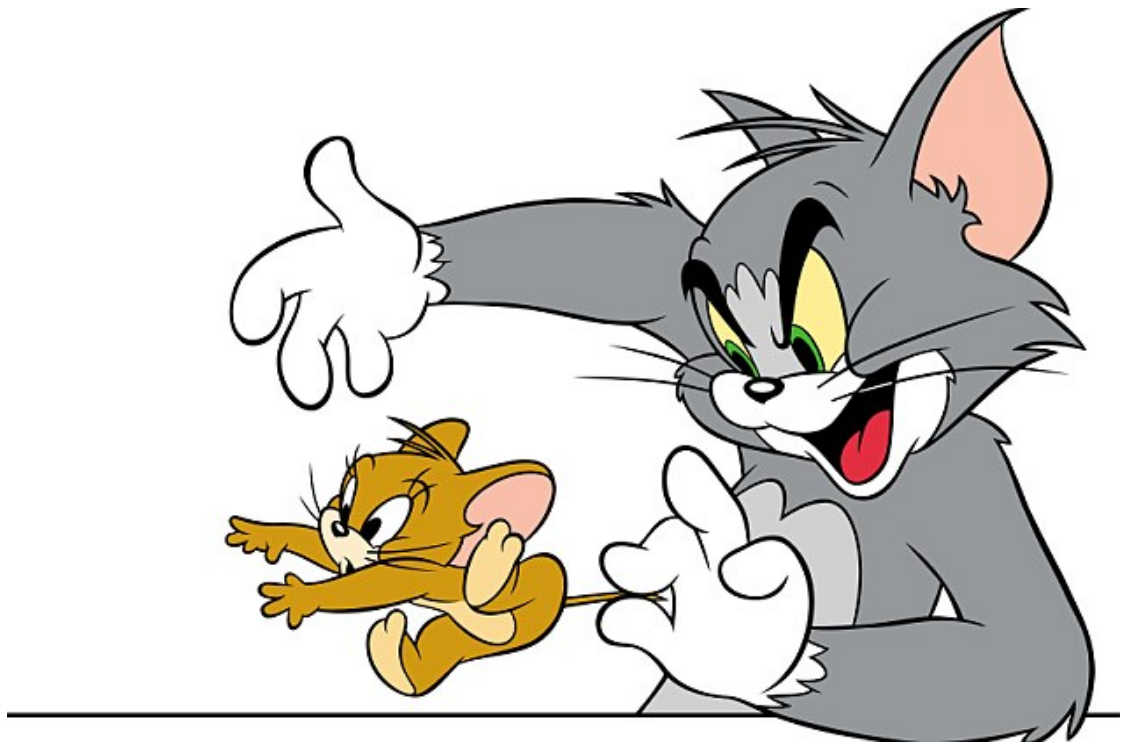


Figure 1.1: A scary cat

helped. It does pose a philosophical problem—we may very well be in the matrix, in which case everything we know is a lie, so our mathematical model could be dead wrong. I will assume without proof that we are not in the matrix. I will assume without proof that logic-checking computers do not lie. I will assume without proof that this bit of circular reasoning we're stuck with is innocent. The proof, of course, is left as an exercise to the reader.

2 boolean logic

This is the simplest interesting model of logic we will discuss. There are two possible values, true and false. When it is not ambiguous, we will abbreviate these as 0 and 1. Letters are variables. There are three operators we start with: and (\wedge), or (\vee), and not (\neg).

Definition 2.0.1 (and \wedge) $x \wedge y$ is true exactly when x and y are both true. This is also called conjunction.

\wedge	0	1
0	0	0
1	0	1

Definition 2.0.2 (or \vee) $x \vee y$ is true exactly when either x or y or both are true. This is also called disjunction.

\vee	0	1
0	0	1
1	1	1

And

Definition 2.0.3 (not \neg) Negation flips 0 and 1:

x	$\neg x$
0	1
1	0

Theorem 2.0.4 (De Morgan's law)

$$\neg(x \wedge y) = (\neg x) \vee (\neg y) \quad (2.1)$$

$$\neg(x \vee y) = (\neg x) \wedge (\neg y) \quad (2.2)$$

Proof. First show eq. (2.1) by checking truth tables:

x	y	$x \wedge y$	$\neg(x \wedge y)$	$\neg x$	$\neg y$	$(\neg x) \vee (\neg y)$
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

□