

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? [\[Wik15b\]](#)

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. Instead of saying, yes it is absolutely true, or no it is not, we will look for algorithms that will check our proofs when possible. 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*.

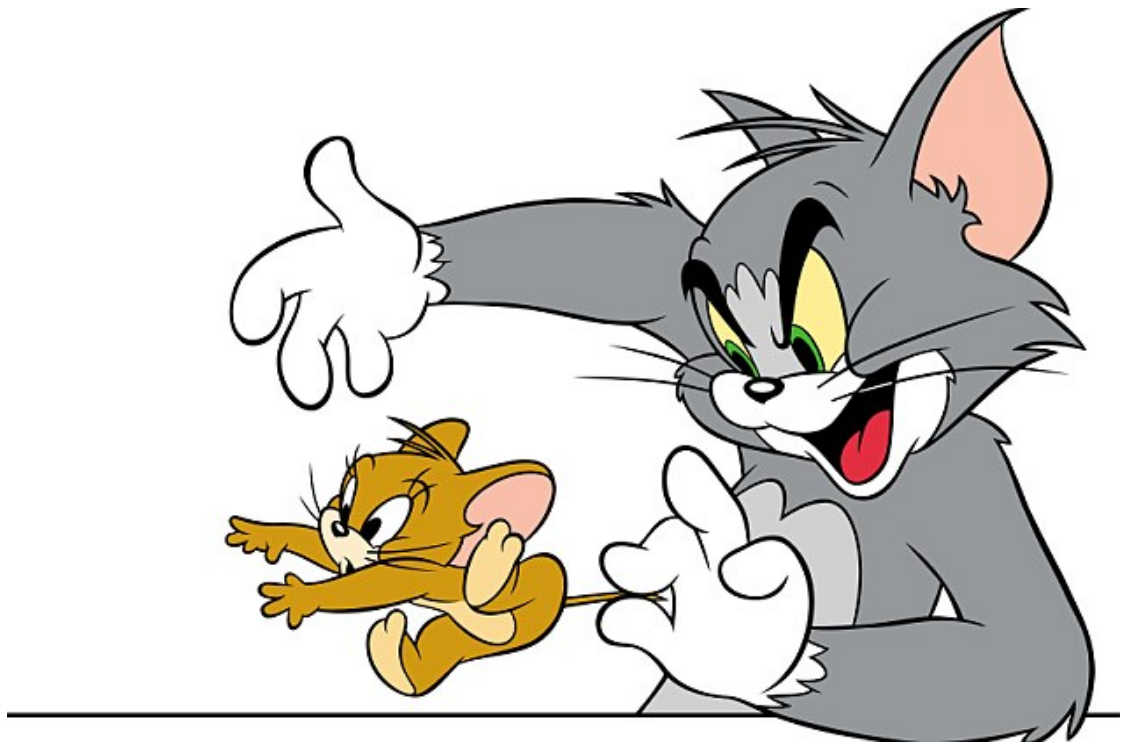


Figure 1.1: A scary cat

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 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.

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)

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

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

(see also [\[Wik15a\]](#))

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

2 Boolean Logic

Then negate both sides of eq. (2.1), giving

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

Make the substitution $\bar{x} = \neg x$ and $\bar{y} = \neg y$:

$$(\neg \bar{x}) \wedge (\neg \bar{y}) = \neg(\bar{x} \vee \bar{y})$$

or, equivalently, eq. (2.2). □

2.1 the conditional

We would like to model causality. Causality is vital in logical reasoning. But defining causality is a black hole ready to swallow up even the cleverest of thinkers. So hark the olde adage ‘the best mathematician is a lazy one’, and punt. We will define the operator implies, a.k.a. the conditional, denoted \rightarrow to act as our stripped-down notion of causality.

Definition 2.1.1 (implies \rightarrow) $x \rightarrow y$ is true for all inputs except $1 \rightarrow 0$.

\rightarrow	0	1
0	1	1
1	0	1

We like this definition because $x \rightarrow y$ captures the idea y must follow from x being true, while allowing for the possibility that y may occur even when x is false. For example, if x means ‘that’s a cat’, and y means ‘that’s a mammal’, $x \rightarrow y$ means ‘that’s a cat, hence it’s a mammal’, with the unspoken caveat that dogs are not cats ($x = 0$), but they are mammals ($y = 1$).

This example is a bit awkward because we have no good way of saying ‘all cats are mammals, but not all mammals are cats’ in boolean logic. We can say this, but not in a particularly useful way. Later we will construct a more sophisticated flavor of logic, predicate logic, that allows such statements.

We defined the conditional through a truth table; however, our definition of boolean logic makes no reference to truth tables. This would suggest we can give an equivalent definition of \rightarrow using \wedge , \vee , and \neg :

Theorem 2.1.2 $x \rightarrow y$ is equivalent to $y \vee (\neg x)$

Proof. Both expressions are false only when $x = 1$ and $y = 0$. □

Definition 2.1.3 (logical equivalence \leftrightarrow) $x \leftrightarrow y$ exactly when $x = y$.

Likewise, we can define this in terms of \wedge , \vee , and \neg :

Theorem 2.1.4 $x \leftrightarrow y$ is equivalent to $(x \wedge y) \vee (\neg x \wedge \neg y)$.

2.2 truth functionality

Definition 2.2.1 (truth function) A map from a string of bits to a single bit is a truth function.

Each truth function has a corresponding truth table, and each truth table defines a truth function. Every boolean expression has a truth table, so every boolean expression is a truth function. But is the reverse true? Can every truth table be expressed in boolean logic?

Theorem 2.2.2 (completeness) Every truth table has a corresponding boolean expression.

Proof. First consider the function $T_{(\alpha_1, \alpha_2, \dots, \alpha_n)}(x_1, x_2, \dots, x_n)$ that is true only when $x_1 = \alpha_1, x_2 = \alpha_2, \dots$. Then let

$$u_i = \begin{cases} x_i, & \text{if } \alpha_i = 1 \\ \neg x_i, & \text{if } \alpha_i = 0 \end{cases}$$

Now we can express $T_{(\alpha_1, \dots, \alpha_n)}$ as

$$u_1 \wedge u_2 \wedge \dots \wedge u_n$$

So, for example, $T_{(1,0,1)}(x_1, x_2, x_3) = x_1 \wedge \neg x_2 \wedge x_3$.

Now consider the arbitrary truth function $f(x_1, \dots, x_n)$. We can write it by or-ing over all $T_{(\alpha_1, \dots, \alpha_n)}$ where $f(\alpha_1, \dots, \alpha_n) = 1$. So for example, the truth table

x_1	x_2	x_3	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

has corresponding expression

$$(\neg x_1 \wedge \neg x_2 \wedge x_3) \vee (x_1 \wedge \neg x_2 \wedge \neg x_3) \vee (x_1 \wedge x_2 \wedge \neg x_3)$$

note, these expressions are uniquely defined by a truth table. □

The resulting expressions in the above proof are said to be in disjunctive normal form.

Corollary 2.2.3 (disjunctive normal form) Any two expressions are equivalent if and only if (iff) they have the same disjunctive normal form.

Proof. Two statements are equivalent iff they have the same truth table. If they have the same truth table, they must have the same disjunctive normal forms. □

This tells us boolean algebra is ‘big enough’ to describe all truth functions. One interesting implication of this result is digital computers can be built.