Contents

1 philosophical underpinnings

Rhetoric could be described as the art of passing off opinions as objective. Through millions of years of evolution, we carbon sack computers have been tuned to take shortcuts. Even in the study of mathematics, these shortcuts are necessary: no one would teach an infant to count by formally proving $2 + 2 = 4$. But with these shortcuts leave us susceptible to logical fallacies, allowing rhetoric to pas off emotions as fact and perceptions 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? [Wik15e]

The solution? A deductive system. One with a well-defined language for clear expression and well-defined rules for clear argumentation. It should be straight-forward enough that proofs can be checked by a computer, so we can claim absolute, or at least unambiguous, truth. To be useful, a deductive system must be expressive enough that we can work out all of mathematics. There is no right balance of these two concerns; each deductive systems balance them differently. They are not at odds with each other, although they seem to describe distinctive worlds.

I will be dodging the question 'is mathematics absolutely true?' In my eyes, a satisfying answer to this question must come from a mathematical analysis. This question is rather resilient to mathematical analysis, thanks to the ambiguity of natural language, the philosophical pitfalls absolutes bring, and even limitations on the mathematical end. Instead, we will pose a series of

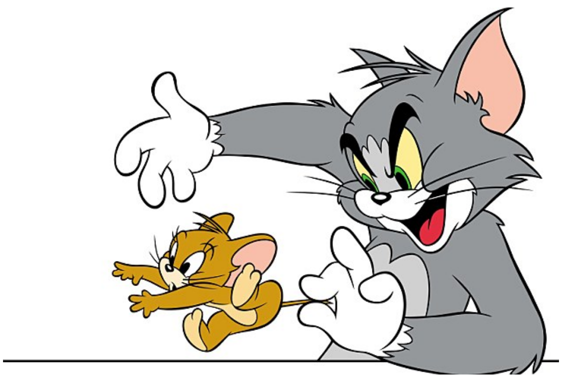

Figure 1.1: A scary cat

mathematical questions that approximate ‘is mathematics absolutely true?’, particularly, ‘can this argument be algorithmically checked?’ and ‘are the valid arguments sensible?’

If we can algorithmically check arguments, we can approach a system of thought immune to subjectivity. This is not to say it is absolutely correct: if we’re wrong, then we’ll all be wrong together. We will elaborate on what sensible means as we go, but a ‘sensible’ argument should be free from contradictions, and should either agree with our intuition, or be able to show us the error in our ways.

How is this done? We will build a model of logical reasoning. With this scaffolding, we will build a model of mathematics. By studying these models, we glean some information about truth in mathematics. Because these models are mathematical, we must ask them mathematical questions.

We’ll need logical to reason about our model. 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 circular reasoning is innocent.

2 boolean logic

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

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)

$$\neg(x \wedge y) = (\neg x) \vee (\neg y) \tag{2.1}$$
$$\neg(x \vee y) = (\neg x) \wedge (\neg y) \tag{2.2}$$

(see also [Wik15b])

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

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 $\bar{y} = \neg y$:

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

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 →)` $x \rightarrow y$ is true for all inputs except $1 \rightarrow 0$.

$\rightarrow$	0	1
0	1	1
1	0	1

This definition of captures the idea y must follow if x is true. But it allows 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 ($y = 1$).

We defined the conditional through a truth table; however, our definition of boolean logic does not mention truth tables. We ought to give a 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 ↔)` $x \leftrightarrow y$ 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 tables to expressions

We have just translated two truth-tables into expressions consisting only of $\wedge$, $\vee$, and $\neg$. You might wonder if we can always do this. We can.

`Theorem 2.2.1 (completeness)` Every truth table has a corresponding boolean expression.

Table 2.1: some truthtable

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

`Proof.` It's best to start with an example. Consider table 2.1 We want an expression that is true exactly when (x_1, x_2, x_3) is $(0, 0, 1)$ or $(1, 0, 0)$ or $(1, 1, 0)$. A straightforward way to do this is to or together a statement that is true only on the input $(0, 0, 1)$, a statement true only on $(1, 0, 0)$, and a statement that is true only on $(1, 1, 0)$.

Consider $(x_1, x_2, x_3) = (0, 0, 1)$. To find an expression that is true for this input and false for all others, we can simply assert $x_1 = 0$ and $x_2 = 0$ and $x_3 = 1$, *i.e.*

$$\neg x_1 \wedge \neg x_2 \wedge x_3$$

By the same argument, we can find the remaining expressions:

$$\begin{aligned} (x_1, x_2, x_3) &= (0, 0, 1) & \neg x_1 \wedge \neg x_2 \wedge x_3 \\ (x_1, x_2, x_3) &= (1, 0, 0) & x_1 \wedge \neg x_2 \wedge \neg x_3 \\ (x_1, x_2, x_3) &= (1, 1, 0) & x_1 \wedge x_2 \wedge \neg x_3 \end{aligned}$$

Hence table 2.1 has expression

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

To prove the theorem, we just need to generalize the work we've done. Suppose we are given a truth table for the function $f(x_1, \dots, x_n)$. Mark off the rows in the truth table where $f(x_1, \dots, x_n) = 1$. For each such row, we construct the expression that asserts the input matches. If the current row is for the input

$$(x_1, \dots, x_n) = (\alpha_1, \dots, \alpha_n)$$

then the expression is

$$\begin{pmatrix} x_1 & \text{if } \alpha_1 = 1 \\ \neg x_1 & \text{if } \alpha_1 = 0 \end{pmatrix} \wedge \cdots \wedge \begin{pmatrix} x_n & \text{if } \alpha_n = 1 \\ \neg x_n & \text{if } \alpha_n = 0 \end{pmatrix}$$

Abbreviate this as $(x_1 = \alpha_1) \wedge \cdots \wedge (x_n = \alpha_n)$. This is just a notational convenience; the final answer will not actually contain $=$ or refer to the α_i.

Or over each of the above expressions:

$$f(x_1, \ldots, x_n) = \bigvee_{\alpha_1,\ldots,\alpha_n : f(\alpha_1,\ldots,\alpha_n)=1} (x_1 = \alpha_1) \wedge \cdots \wedge (x_n = \alpha_n)$$

giving the desired expression. □

This tells us that boolean logic is big enough to express any truth table. An interesting implication is that general purpose computers can be constructed!

2.3 absolute truth?

Boolean logic is simple enough that we can talk about absolute truth without making a mess. Trusting boolean algebra is about as reasonable as trusting a calculator for integer arithmetic. A boolean expression's truth value can be computed directly from the inputs. Since each input can be either true or false, there are only finitely many possible inputs. Everything you need to know is contained in the expression's truth table. These computations can be carried out unambiguously by a machine in finite time, so things are looking pretty good.

A statement and its negation cannot be true simultaneously, *i.e.*

$$p \wedge \neg p = 0$$

for any expression p. No matter what p is, it must evaluate to either 1 or 0. Suppose $p = 1$. Then $\neg p = 0$. So $p \wedge \neg p = 1 \wedge 0 = 0$. Likewise, if $p = 0$, we get $0 \wedge 1 = 0$.

3 predicate logic

Boolean logic is beautiful, but it is not expressive enough to make deductions like 'Socrates is a man. All men are mortal. Therefore, Socrates is mortal.'

Nineteenth century mathematics was in crisis. For thousands of years, mathematics had remained essentially unchanged: the gold standard was proof by Euclidean geometry. Algebra, arabic numerals, zero were tools to expedite geometric reasoning. Even new results in number theory were assumed to be, at heart, geometric truths. But by the 1800s, it was clear this worldview was no longer sustainable. Mathematics had grown too abstract.

Calculus, the most revolutionary discovery this side of the dark ages, stood with embarrassingly poor theoretical footing. Newton and Leibniz through Euler and Cauchy relied on infinitesimals, but could not justify their use *a priori*, certainly not by Euclidean geometry. To make matters worse, if you were not careful, an argument that seemed kosher could result in nonsense. Fourier's work was downright scandalous. He was able to solve the heat equation, a problem unsolved for all but the most trivial cases. Experiments verified his results, but conservative mathematicians refused to accept his method.

Abel, and later Galois, proved that degree 5 equations have no general solution that can be written in terms of the basic arithmetic operations and radicals. In particular, the crux of Galois's proof were abstract systems with no clear base in Euclidean geometry.

Even Euclidean geometry was proving to be less than the paragon of flawless logic it once seemed. Problems in Euclidean geometry went unsolved for millenia, like can you trisect an angle with a compass and straightedge went unsolved for a few millennia. But these were quickly solved using Galois's tools.[Wik15a] Euclid's parallel axiom was thought to be provable from his other axioms for thousands of years, but it was shown in the mid 19th century, that it was in fact independent, as Euclid's remaining axioms are satisfied by spherical and hyperbolic geometry, in both of which Euclid's parallel postulate does not hold.

In the nineteenth century, mathematics outgrew its ontology. The situation was so disorienting, Dodgson wrote a veiled satire ridiculing the absurdity of this new mathematics. Dodgson is better known as Lewis Carol. This satire is called *Alice in Wonderland*.[Ang15]

Gottlob Frege laid the foundations for mathematics' new ontology. Whereas George Boole's logic was viewed more as a useful trick that cleaned up some ambiguities for the pedants, Frege developed a logic capable of housing all of the new mathematical objects. His work went relatively unnoticed by his contemporaries, but it revolutionized mathematics and philosophy.

3.1 what is predicate logic?

First order predicate logic is the modern descendant of Frege's logic. It is more expressive than boolean logic, making it both critical to our modern understanding of mathematics, and complicated enough that we can't declare it absolutely true.

It makes use of the same operations and conventions of boolean logic ($\wedge, \vee, \neg, \rightarrow, \leftrightarrow$ and variables), but adds constant symbols, function symbols, predicate symbols, and quantifiers, and a special equality predicate.

These additions allow us to talk in generalities about entities that live in some universe, for example the natural numbers.

`Definition 3.1.1 (constant symbols)` Constant symbols name specific entities in the universe.

`Example 3.1.2 (constant symbols)` $0, 1, 30, 10$ are constant symbols referring to natural numbers.

`Definition 3.1.3` Function symbols name functions that take in entities and output an entity.

The arity of a function is the number of arguments it takes, *e.g.* $+$ has arity 2. Functions do not reside in the proverbial universe. If we are working in a universe of natural numbers, we will want to refer to $+$. But $+$ is not a number. It belongs in a separate bin.

`Example 3.1.4` $+, \times$ are functions of natural numbers.

`Definition 3.1.5 (predicate symbols)` Predicate symbols name predicts, which take in entities and output true or false. Like functions, predicates have arity and exist outside of the universe. But predicates are distinct from functions. Predicates output truth values, but functions output elements of the universe (*e.g.* natural numbers). If add two numbers, you can multiply the result by 2: $(1+3) \times 2$. You cannot however, say $(1 < 3) + 5$. We need a separate bin for $<$.

`Example 3.1.6` $<, \geqslant, =$ are predicates of natural numbers

`Definition 3.1.7 (quantifier)` Quantifiers allow us to talk about general elements in the universe. There are two quantifiers: $\forall$, read as 'for every', and $\exists$, read as 'there exists'.

Intuitively, $\forall x \phi(x)$ means $\phi(x)$ is true, no matter what x is in the universe, and $\exists x \phi(x)$ means somewhere in the universe, there is an x that satisfies ϕ. We will make this exact through inference rules later.

If we refer to Socrates by the constant symbol s, we can state the syllogism 'Socrates is a man. All men are mortal. Hence, Socrates is mortal' as follows:

$$\texttt{man?}(s) \wedge \forall x \big(\texttt{man?}(x) \rightarrow \texttt{mortal?}(x)\big) \rightarrow \texttt{mortal?}(s)$$

Recall the statement

> Every mouse fears some cat

We can now state this unambiguously as

$$\forall m \big(\texttt{mouse?}(m) \rightarrow \exists c(\texttt{cat?}(c) \wedge \texttt{fears?}(m, c))\big)$$

i.e. 'for every mouse, we can find a cat which that specific mouse fears.'

$$\exists c \big(\texttt{cat?}(c) \wedge \forall m(\texttt{mouse?}(m)\texttt{fears?}(m, c)\big)$$

i.e. 'there is one particularly frightening cat all mice fear'.

`Definition 3.1.8 (equality)` Equality is a predicate that always satisfies the following axioms

1. Reflexivity: $\forall x(x = x)$

2. Substitution: equal elements are interchangeable

   for all functions f: $\forall x \forall y(x = y \to f(x) = f(y))$[1]

   for predicates ϕ: $\forall x \forall y(x = y \to \phi(x) \leftrightarrow \phi(y))$

3.2 peano arithmetic

I promised this would be sufficient to house modern mathematics. I will show you how the natural numbers can be defined using Peano's axioms [Wik15d].

`Definition 3.2.1 (Peano arithmetic)` There is a constant called 0 and a function of one argument called succ. The Peano axioms are as follows:

1. Nothing comes before zero:
$$\forall x. 0 \neq \mathsf{succ}(x)$$

2. Two naturals are equal if their successors are equal:
$$\forall x \forall y. \big(\mathsf{succ}(x) = \mathsf{succ}(y) \to x = y\big)$$

3. Induction: if Φ is a predicate,
$$\Phi(0) \wedge \forall x\big(\Phi(x) \to \Phi(\mathsf{succ}(x))\big) \to \forall y. \Phi(y)$$

   i.e., if $\Phi(0)$ and $\Phi(s)$ implies $\Phi(s+1)$, then Φ must be true for all natural numbers.

`Definition 3.2.2 (plus)` We can define $+$, a function of two inputs, by recursion:

$$\forall x(x + 0 = x)$$
$$\forall x \forall y\big(x + \mathsf{succ}(y) = \mathsf{succ}(x + y)\big)$$

[1] Quantifiers range over the universe. But functions don't live in the universe. So we can't write something like $\forall f \forall x \forall y(x = y \to f(x) = f(y))$. The solution is to treat this condition as a template, or *axiom schema*, and copy it over for each function.

3.3 deduction

In boolean algebra, when determining the truth of a statement, the worst case scenario will require writing out a large truth table. Not so for predicate logic. Quantifiers range over everything in some possibly infinite universe. Predicates abstract away calculations to something behind the scenes. Without any way to explicitly enumerate everything in the universe, the logician must indicate its content, as well as the behavior of some predicates, with axioms.

Some axioms, like $\forall x P(x)$ and $\exists x \neg P(x)$ are contradictory, hence nonsensical. One would hope this is just a case of 'ask a bad question; get a bad answer'. We would hope that the laws of deduction, which govern what you can actually do with the axioms you choose, are ok.

For $\wedge, \vee, \neg, \rightarrow, \leftrightarrow$, deduction works the same as it did for boolean logic:

$x \wedge y$ is true if and only if (iff) x is true and y is true

$x \vee y$ is true iff x is true or y is true

$\neg x$ is true iff x is false

$x \rightarrow y$ is true unless $x = 1$ and $y = 0$

$x \leftrightarrow y$ is true iff x has the same truth value as y

Universal Introduction: If we know $\phi(\alpha)$, we can conclude $\forall \alpha . \phi(\alpha)$ so long as α is unused elsewhere.

Universal Elimination: If we know $\forall \alpha . \phi(\alpha)$ then we can conclude $\phi(\beta)$.

Existential Introduction: If we know $\phi(\beta)$ then we can conclude $\exists \alpha \phi(\alpha)$ so long as α is previously unused.

Existential Elimination: If we know $\exists \alpha \phi(\alpha)$, we can say $\phi(\beta)$ so long as β is previously unused.

Applications of some rules may require renaming variables. This is no problem as long as the meaning of the statement is not changed (*i.e.* two things that used to have different names cannot have the same name).

3.4 some proofs

We are now prepared to prove things about Peano arithmetic:

`Theorem 3.4.1 (+ is associative)` The statement

$$\forall x \forall y \forall z . (x + y) + z = x + (y + z)$$

is deducible from Peano arithmetic and the definition of plus.

`Proof.` Short version—this is a proof by induction. First, we prove the base case:

$$\forall a \forall b . (a + b) + 0 = a + (b + 0) \tag{3.1}$$

Equivalently, $\phi(0)$ where

$$\phi(x) = \forall a \forall b.(a+b)+x = a+(b+x)$$

Noting that $y+0=y$, (see the definition of plus), we see eq. (3.1) is, in fact, true.

Now we prove the inductive step:

$$\forall x\big(\phi(x) \to \phi(\mathsf{succ}(x))\big) \tag{3.2}$$

Pick a, b, and c. Assume $(a+b)+c = a+(b+c)$, which we abbreviate as $a+b+c$. Now consider $(a+b)+\mathsf{succ}(c)$. By definition of plus, we see

$$(a+b)+\mathsf{succ}(c) = \mathsf{succ}(a+b+c)$$

Similarly,

$$a+(b+\mathsf{succ}(c)) = a+\mathsf{succ}(b+c) = \mathsf{succ}(a+b+c)$$

hence $(a+b)+\mathsf{succ}(c) = a+(b+\mathsf{succ}(c))$. This proves the inductive step, eq. (3.2).

To finish the proof, use the induction axiom. □

The above proof does not justify each step, but gives enough information to show how deduction works. This is the norm for mathematical proofs. Actually, such a proof is boring when you are studying the mathematics of the natural numbers: it is not hard to convince someone addition is commutative.

But since we want to understand the very fabric of reason itself, it must be possible to write out this proof in such painstaking detail that a computer, or even the most pedantic of logic professors, can follow it. Such a proof may look like the following:

`Proof.` Full gory details—First establish the base case, eq. (3.1) Consider the statement $(a+b)+0$. By definition 3.2.2, we know $\forall x.x+0 = x$. By universal elimination, deduce $(a+b)+0 = a+b$. Now consider $a+(b+0)$. By universal elimination again, deduce $b+0=b$. By substitution, deduce $a+(b+0) = a+b$. Then, by transitivity, we know $(a+b)+0 = a+(b+0)$. As a and b are totally arbitrary free variables, two universal introductions establish eq. (3.1).

Now the inductive step, eq. (3.2) Assume

$$(a+b)+c = a+(b+c) = a+b+c \tag{3.3}$$

Consider the statement $(a+b)+\mathsf{succ}(c)$. From definition 3.2.2, we know

$$\forall x \forall y.x+\mathsf{succ}(y) = \mathsf{succ}(x+y)$$

An analogous quantifier elimination gives $(a+b)+\mathsf{succ}(c) = \mathsf{succ}(a+b+c)$. Another quantifier elimination gives $a+(b+\mathsf{succ}(c)) = a+\mathsf{succ}(b+c)$. A final quantifier elimination gives $a+\mathsf{succ}(b+c) = \mathsf{succ}(a+b+c)$. Transitivity gives $(a+b)+\mathsf{succ}(c) = a+(b+\mathsf{succ}(c))$. We can stop assuming eq. (3.3), rephrasing our work so far as

$$(a+b)+c = a+(b+c) \to (a+b)+\mathsf{succ}(c) = a+(b+\mathsf{succ}(c))$$

Now a, b, and c are arbitrary, so we can use a triple quantifier introduction to prove eq. (3.2).
As we have shown eqs. (3.1) and (3.2) we can deduce

$$\phi(0) \wedge \forall x\big(\phi(x) \to \phi(\mathsf{succ}(x))\big) \tag{3.4}$$

By the induction axiom,

$$\phi(0) \wedge \forall x\big(\phi(x) \to \phi(\mathsf{succ}(x))\big) \to \forall x \phi(x)$$

Then using boolean logic (*modus ponens*), conclude $\forall z \phi(z)$. Hence, $+$ is associative. This proves the theorem. □

Ironically, justifying each step results in a proof that is harder for a human to understand.

3.5 the death of reason

Once mathematicians started actually reading Frege, it was not long until they would dream the impossible dream. Hilbert and his entourage longed for a world in which mathematics had been effectively solved, the right axioms were chosen, and a machine would be built that would write out every true theorem until the end of time. All mathematicians would then be free to sip martinis poolside as an endless tickertape listing all true things would scroll by. [Wik15c]

The mathematicians would have to really read this carefully every now and then to answer a physicist's question. But they discovered by speaking in gratuitous generalities and shouting 'the proof is trivial' rather frequently, they could make a pesky physicist disappear, ensuring a veritable utopia (see `http://www.smbc-comics.com/index.php?db=comics&id=2675#comic`). Thus French Formalism was born.

But, one fateful night in 1931, dear reader, a young Kurt Gödel dealt a fatal blow to the Hilbertian dream of mathematical rapture. He was merely engaging in the logician's favorite pass-time: making things mercilessly meta. As many a logician and patron of webcomics know, making things meta may make a mess, and a major mess it made moreover. To his merit, this metamethematical misfortune marks the makings of a modern mathematical maturity, one more meaningful than a mere mirage marred by the multitude of mathematical models and their unmarried meandering truths. Any system of mathematics sufficiently sophisticated to serve as semantics for something like Peano's arithmetic suffers from a malady said to be incompleteness. See, m'lady, such a malady means more than several true-seeming statements may lay beyond reach of the most surefire machines of Hilbert's dreams.

3.5.1 berry's paradox

There are only so many numbers you can name in under eleven words, finitely many in fact. So consider the smallest positive integer not nameable in under eleven words. Surely it exists; every nonempty set of naturals has a least number (exercise for the reader!). When considering how many words it takes to name this number, 10 is right out. Except, of course, I named it in 10 words: 'the(1) smallest(2) positive(3) integer(4) not(5) nameable(6) in(7) under(8) eleven(9) words(10)'.

3.5.2 incompleteness

This proof is due to George Boolos. Gödel's proof used the liar's paradox (this sentence is false), which leads to a longer proof.[Wik15f] In both cases, the genius of the proof lies in encoding these paradoxes into Peano arithmetic.

Definition 3.5.1 (Gödel numbering) A Gödel numbering is a lossless encoding of first order logical statements in the natural numbers. Many such encodings exist. For our purposes, any one will work.

Once we've picked a Gödel numbering, Peano arithmetic becomes self-referential. We can talk about the theory of predicate logic as if it coexisted in the same universe as the natural numbers themselves.

Definition 3.5.2 (names) A formula ϕ names the number n iff $\forall x(F(x) \leftrightarrow x = n)$.

Now we define some predicates:

Definition 3.5.3 (nameln?) $\mathtt{nameln?}(x, y)$ is true iff x can be named by a statement containing y symbols. Via the Gödel numbering, this can be defined using only the alphabet of peano arithmetic.

Definition 3.5.4 (smaller?)

$$\mathtt{smaller?}(x, y) \leftrightarrow \exists z. z < y \wedge \mathtt{nameln?}(x, z)$$

i.e. $\mathtt{smaller?}(x, y)$ iff x can be named by a statement with length less than y.

Definition 3.5.5 (nodef?)

$$\mathtt{nodef?}(x, y) \leftrightarrow \neg\mathtt{smaller?}(x, y) \wedge \forall a\big(a < x \rightarrow \mathtt{smaller?}(a, y)\big)$$

i.e. $\mathtt{nodef?}(x, y)$ iff x is the smallest number that cannot be named in y symbols.

Definition 3.5.6 (berry?)

$$\mathtt{berry?}(x) \leftrightarrow \exists y. y = \mathtt{succ}^{10}(0) \times \mathtt{succ}^{k}(0) \wedge \mathtt{nodef?}(x, y)$$

where $\times$ is multiplication, $\mathtt{succ}^n$ is n applications of $\mathtt{succ}$, and k is the length of $\mathtt{nodef?}(x, y)$. This encodes Berry's paradox in Peano arithmetic.

Theorem 3.5.7 (Gödel's first incompleteness) There is some statement that cannot be proven true or false in Peano arithmetic.

Proof. We will construct such a statement, called a Gödel sentence. Let N be the smallest number that cannot be named in fewer than $10k$ symbols. The sentence

$$\forall x\big(\mathtt{berry?}(x) \leftrightarrow x = \mathtt{succ}^N(0)\big) \tag{3.5}$$

is a Gödel sentence. A proof of its negation would contradict the fact that N is the smallest number that cannot be named in fewer than $10k$ symbols. A proof of eq. (3.5) would imply $\mathtt{berry?}$ names N in fewer than $10k$ symbols. □

Bibliography

[Ang15] Devlin's Angle. *The hidden math behind* Alice in Wonderland. 2015. URL: https://www.maa.org/external_archive/devlin/devlin_03_10.html.

[Wik15a] Wikipedia. *Angle trisection*. 2015. URL: https://en.wikipedia.org/wiki/Angle_trisection.

[Wik15b] Wikipedia. *De Morgan's laws*. 2015. URL: https://en.wikipedia.org/wiki/De_Morgan%27s_laws.

[Wik15c] Wikipedia. *Hilbert's Program*. 2015. URL: https://en.wikipedia.org/wiki/Hilbert%27s_program.

[Wik15d] Wikipedia. *Peano axioms*. 2015. URL: https://en.wikipedia.org/wiki/Peano_axioms.

[Wik15e] Wikipedia. *Problem of multiple generality*. 2015. URL: https://en.wikipedia.org/wiki/Problem_of_multiple_generality.

[Wik15f] Wikipedia. *Proof sketch for Gödel's first incompleteness theorem*. 2015. URL: https://en.wikipedia.org/wiki/Proof_sketch_for_G%C3%B6del%27s_first_incompleteness_theorem.