On Formal Logic

Johannes Siedersleben
Oxford, August 2024

1. Six men who shaped the world of logic
2. Mathematics is an abstraction of our world
3. Formal logic is an abstraction of mathematics
4. Self-referential contradictions
5. Three celebrated theorems
6. The Curry-Howard Correspondence
7. The Big Picture: A Snapshot of 2500 Years of Evolution
8. More theories
9. Some books

Six men who shaped the world of logic

Alan Turing, 1912 - 1954

Haskell Curry, 1906 - 1978

Kurt Gödel, 1906 - 1978

Alonso Church, 1903 - 1995

William A. Howard, 1926-

Alfred Tarski, 1901 - 1983

Turing

The Turing Medal is to computer scientists what the Nobel Prize is to other disciplines.
Movie: The Imitation Game (2014)
Turing was gay and was forced to undergo chemical treatment. He died by suicide aged 42.

His achievements:
He invented the Turing machine, a model (abstraction) of a computer, before computers existed.
He invented a programming language for this hypothetical computer.
He wrote an interpreter for that language, using that language itself.
He answered the decision problem posed by Hilbert (1928) in the negative. He used the same techniques as Gödel.
He proved the equivalence of the Turing machine and Church's lambda calculus.
He invented the Turing test (how to tell a computer from a human)
He was instrumental in breaking the Nazi code during the Second World War, helping to end submarine warfare in the North Atlantic.
He ran the marathon in about 2:40.

Look up the others on Wikipedia.

Mathematics is an Abstraction of Our World and Much More

Mathematics
(Coherence)

Arithmetic
(numbers; axioms, deduction rules)

Euclidean geometry
(circles, lines; axioms, deduction rules)

(Pragmatism)

abstract

apply

Our world
(common sense)

countable objects
(apples, oranges)

regular shapes
(wheels, poles)

Mathematics is an Abstraction of Our World and Much More

I make no assumptions about our world. What I am assuming is that humans are somehow capable of abstraction: from countable objects to numbers, from regular shapes to lines, planes, and circles and that they are somehow capable of transferring the results back to our world.

The box on the left represents mathematics as you would find it in any textbook. I am not dealing with questions like "Are numbers invented or found" or "Do numbers exist independently of people". All this is interesting, but not relevant to my story.

Abstractions take us out of our world and into the world of mathematics. Mathematics is a bubble of coherent truth. If you enter it, you accept the rules of the game, the way mathematical theorems and proofs must be. Applying mathematics, i.e. transferring the results back to our world, amounts to a pragmatic attitude to truth: true is what works.

In mathematics, say arithmetic, we develop theorems and algorithms. We deliberately ignore what is being counted. We return to our world by applying theorems and algorithms to real objects. When you add numbers for fun or practice, you are doing mathematics; when you check your bank account by adding pounds, you are applying mathematics. Why does $1 + 1 = 2$? One of the axioms of arithmetic says that every natural number has exactly one successor (that's what '+1' means), and the successor of 1 is called 2. So, saying that $1 + 1 = 2$ is like saying that a triangle has 3 angles.

Axioms can be necessary or redundant, useful or not, but they can never be true or false. An axiom is a statement whose truth is taken for granted in a given system. It has taken a long time to arrive at the system of axioms we now have.

Theorems and algorithms can be true (or false) in two ways: They can be proved by mathematical standards (that would be coherence) and/or they can be tested against our world by counting concrete objects, such as apples. In the latter case, the truth is pragmatic (useful) or by correspondence (matching our world).

There is more to mathematics than its application to our world. Many theorems are interesting in themselves (Fermat's Last Theorem) with no prospect of ever being applied, others lie around for decades and suddenly take on unexpected importance.

Three Benefits of Abstraction

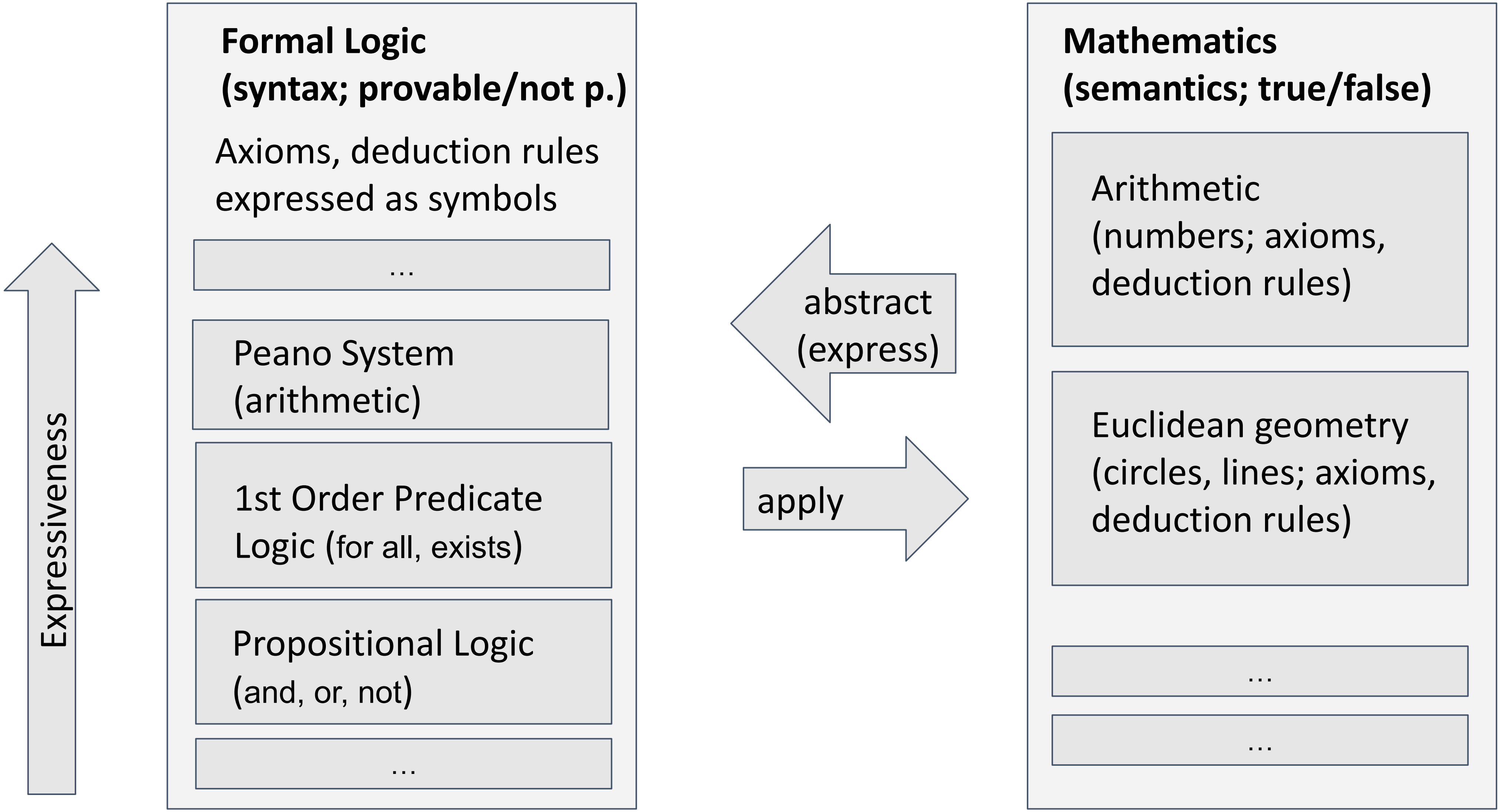
Efficiency by abstracting once, applying many times

Insight by jettisoning information for better understanding

Correctness: what has been abstracted away cannot creep back in as a hidden assumption.

What a paradox to gain insight by throwing information overboard!
But doing mathematics with apples and oranges is like running 100 metres flat with a backpack.

Formal Logic is to Mathematics what Mathematics is to Our World.



Formal Logic is to Mathematics what Mathematics is to Our World.

Claudius: My words go up, my thoughts remain below, words without thoughts never to heaven go.
Words without thoughts don't go to heaven, but symbols without meanings are very useful!

Here is the main message of my talk: Formal logic is to mathematics what mathematics is to our world.
Formal logic is an abstraction of mathematics, and mathematics is an application of logic.

But what is being abstracted? It is the meaning of the symbols! Symbols are used in mathematics. In arithmetic, they represent numbers, in geometry, they represent circles, lines, and planes. In formal logic you use symbols which have no meaning. What you have is a grammar that defines exactly what terms and propositions there are, you have a list of axioms (true propositions), and a list of rules of deduction (very few, often just the modus ponens).

What is a grammar? A grammar tells you that e.g. "a + b - c" is syntactically correct, whereas "++ab-" is not.

In formal logic there are many alternative, sometimes competing, formal systems, such as PL0 (propositional logic), PL1 (first-order predicate logic), PA (Peano arithmetic), to name but a few. Each of these systems has its own grammar, axioms, and rules of deduction. Reasoning within such a system is strictly reduced to symbol substitution (e.g. modus ponens: if A and A => B, then B). A proposition such as "a + b = b + a" is provable if it can be derived from the given axioms by a finite sequence of symbol substitutions. In a formal system, true and false are meaningless. A proposition is either provable or not.

Chess can be thought of as a small formal system. The true propositions are all states of a chessboard that can be reached by legal moves, and these moves are the proof. It is easy to write a program that generates all possible chess games (a huge number with over 120 digits), and you can do this for any formal system. In the case of chess the programme would run for many billions of years (chess is finite), and in general it would run forever (most formal systems are infinite)

Historically, systems like propositional logic or predicate logic have been around for a long time. Propositional logic goes back to Aristotle.
It was Gödel and others who reduced logic to mere meaningless symbols.

Insights: Self-Referential Contradictions

(This theorem holds under weak assumptions)

Gödel's Diagonalization Lemma

In most systems, there are propositions that contain a contradiction in themselves.

Some funny examples

"Is a palindrome" is not a palindrome.

"Is misspelled" is spelled correctly.

"Sounds sophisticated" doesn't sound sophisticated.

Two serious ones

"This is not provable" is provable.

"This is false" is true.

Insights: Three Celebrated Theorems

A system **S** is called **consistent**, if there is no proposition **p** such that **p** and **not p** are provable (“ex falso quodlibet”).
S is called **decidable**, if, for each proposition **p**, either **p** or **not p** is provable.

(the following theorems hold under weak assumptions)
Gödel’s first incompleteness theorem
If a system is consistent, it is not decidable

Tarski’s theorem
It is impossible to define “truth” within a consistent system

Gödel’s second incompleteness theorem:
The consistency of a system cannot be proven within the system itself.

Three Celebrated Theorems

A lot can go wrong in formal systems.

A crucial property is consistency. In an inconsistent system you have at least one proposition p such that p and not p are provable.
In this case, the system is worthless, because everything follows from the absurd: If I were the emperor of China, I could promise everything.

A desirable property is decidability, i.e. for any property p you can prove p or not p.
An inconsistent system is always decidable (you can prove everything); in a consistent and decidable system, the or is exclusive.

The first incompleteness theorem and Tarski's theorem follow directly from the diagonalisation lemma.
They show us the limits of what we can achieve in formal systems.

The smallest formal system for which these theorems hold is Robinson arithmetic, i.e. Peano arithmetic without induction.

The second incompleteness theorem is devastating. It shows that for most important systems there will never be a proof of consistency, which is so essential!

The largest formal system whose consistency can be proved is Presburger arithmetic, i.e. Peano arithmetic without multiplication.

There is one huge difficulty Gödel had to overcome: He had to find a way how to translate normal mathematics into formal logic.
This is much trickier than you might think. It involves Gödel numbering and primitive recursive functions (whatever that means).
The proof is over ten pages and very hard to read, but the key idea is simple.

Efficiency and Insight: The Curry-Howard Correspondence

Programs
(syntax; programs terminating or not)

Data Types

Programs

Curry-Howard

Formal Logic
(syntax; propositions provable or not)

Theorems

Proofs

The Curry-Howard Correspondence

The theory of programming was developed independently by Church and Turing.
Church came up with the lambda calculus, a very abstract but powerful programming language.
Turing came up with his Turing machine. His paper appeared in 1936, four weeks after Church's.
Turing, slightly stressed, showed the equivalence of the two approaches within a few days. Both authors had a negative answer to the famous decision problem given by Hilbert in 1928:

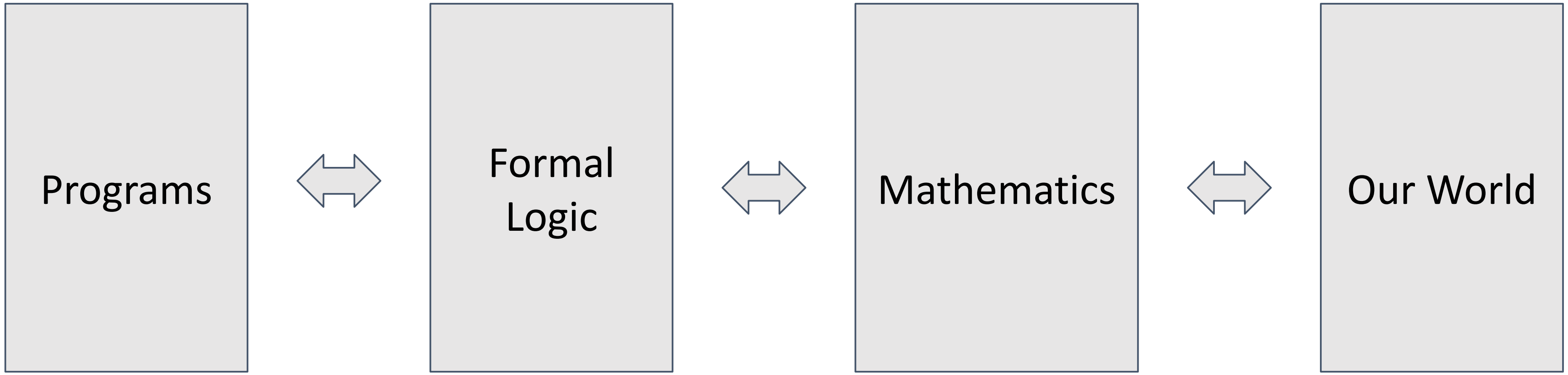
“Given a formal system with a set of axioms and rules of inference, is there a mechanical procedure (algorithm) that, for any given statement in the system, can determine whether that statement is provable (i.e., a theorem) within the system?”

One of the most amazing results of all time is the Curry-Howard correspondence between logic and computer science. It says:

There is a one-to-one relationship between theorems and data types (static structures of programming languages), and another between proofs and (terminating) programs.
Every theorem can be mechanically translated into a data type and vice versa.
Any proof can be mechanically translated into a terminating programme and vice versa.

So formal logic and computing are two sides of the same coin, and, at least in theory, mathematical proofs are proved correct when the corresponding programme terminates (which can take a while). On the other hand, all incompleteness results apply to both sides: There are many problems that can never be programmed.

The Big Picture:
A Snapshot of 2500 Years of Evolution



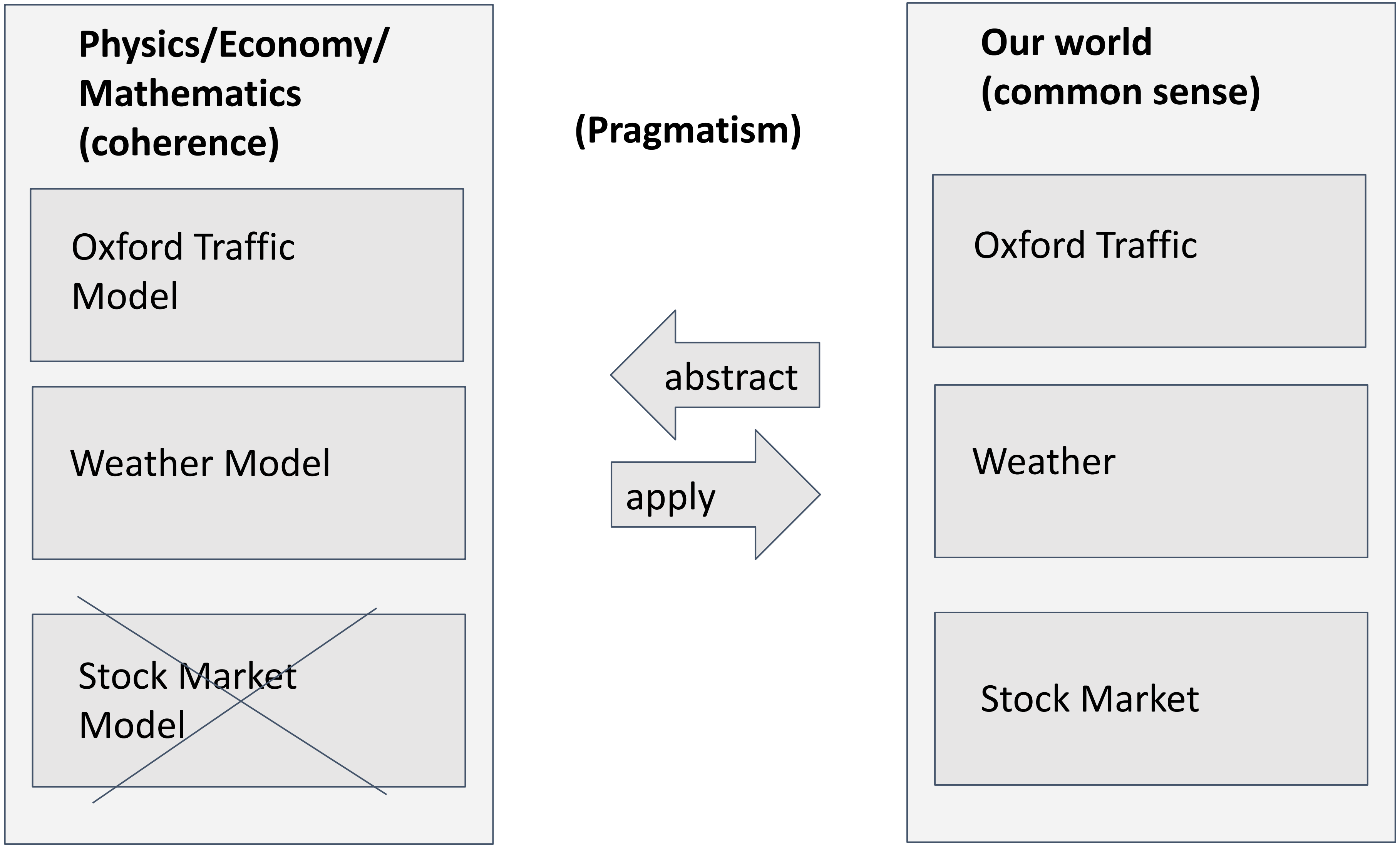
A Snapshot of 2500 Years of Evolution

I look at the big picture in awe. It is another "Greatest Show on Earth".
It shows the deep relationship between mathematics, logic, and computer science. Why is it the way it is?

Its roots go back to Aristotle, but most of it was created between 1900 and 1970. So what we see is a snapshot: 1900, 1930, 1950 would have seen as many pictures. What will it look like in 50 or 100 years? There is a lot of work to be done; there are countless problems to be solved in all areas.

Can there be any kind of necessity? Of course, 1 + 1 = 2 will always be true, but that is not the point. Are we condemned to live forever with these disturbing results of incompleteness? Could there be something completely new, such as quantum logic, that opens up new horizons? If I knew, I would tell you.

More Theories (or Models)



More Theories (or Models)

Mathematical theories are, in a sense, just abstractions of reality. We have all heard of models. Here are three examples, two of which are very successful: we can model (or simulate) traffic and the weather, but for some reason we cannot model the stock market.

The principle of modelling traffic is exactly the same as we have seen in mathematics: You abstract away the irrelevant, and you end up with a model of (say) Oxford that includes the roads, the bus lanes, the traffic lights and so on. This model exists as a computer program. So you can simulate rush hour, you can simulate the effect of roadworks, or you can play with traffic light control. This kind of modelling is very useful and not controversial.

Meteorologists have developed huge weather models that run on supercomputers. The models in use today are the survivors of a long Darwinian competition; the IPCC’s predictions about climate change are the results of these models. They can be tested against the past: given data from, say, the eighties, they predict the nineties. Depending on the quality of the prediction, you believe (or not) the predictions for the coming decades. The competition among models is normal business; this is how science works and why it is successful. This whole business represents another bubble of coherence: Those who participate accept the rules of the game. Climate deniers do not.

I use the stock market as another example because nature has conspired against modelling it. No attempt to model it has ever succeeded, and if it did, the stock market would crash immediately.

Some Books

There is no easy way to formal logic. [5] is not as easy as many think, and it only introduces toy systems rather than the real thing.
[1] and [6] are for experts only, [3], [4], [7], [8], [9], [10] for dedicated students willing to work hard. [2] gives a good overview if you know the basics. I find [9] quite readable.

1. H.-D. Ebbinghaus, J. Flum, W. Thomas: Einführung in die mathematische Logik. Spektrum Akademischer Verlag, 1998

2. Torkel Franzén: Gödel's Theorem. An Incomplete Guide to its Use and Abuse. CRC Press, 2005

3. Daniel Friedman, David Christiansen: The Little Typer. MIT Press, 2018

4. Dirk Hoffmann: Grenzen der Mathematik. Springer, 2013

5. Douglas Hofstadter: Gödel, Escher Bach. Basic Books, 1999

6. Kurt Gödel: On formally undecidable propositions of Principia Mathematica and related systems.
   Dover Publications (first published 1931)

7. Peter Smith: A introduction to Gödel's theorems. Cambridge University Press, 2007

8. Peter Smith: A introduction to formal logic. Cambridge University Press, 2020

9. John Stillwell: Roads to Infinity. CRC Press, 2010

10. Alfred Tarski: Einführung in die mathematische Logik. Vandenhoeck & Ruprecht, 1977