Lectures on Philosophy and Computation

Cristian S. Calude

Semester 1, 2019

Hilbert's programme

Big data and AI

Sans les mathématiques on ne pénètre point au fond de la philosophie.
Sans la philosophie on ne pénètre point au fond des mathématiques.
Sans les deux on ne pénètre point au fond de rien.

Without mathematics one cannot understand the fundamentals of philosophy.
Without philosophy we cannot reach the foundation of mathematics.
Without both (mathematics and philosophy) one cannot reach anything that is fundamental.

Russell's paradox, discovered by Bertrand Russell in 1901, showed that the naive set theory created by Georg Cantor leads to a contradiction. This generated a crisis in the foundations of mathematics.

According to naive set theory, any definable collection is a set.

Let R be the set of all sets that are not members of themselves.

If R is a member of itself, it would contradict its own definition as the set containing all sets that are not members of themselves.

If R is not a member of itself, it would qualify as a member of itself by the same definition, again a contradiction.

Give examples of:

1. sets which are not members of themselves,
2. sets which are members of themselves.

The main goal of Hilbert's program was to provide secure foundations for all mathematics. This includes:

- A formalisation of mathematics: all mathematical statements should be written in a precise formal language, and manipulated according to well defined rules (formal system).
- Completeness: a proof that all true mathematical statements can be proved in the adopted formal system.
- Consistency: a proof—preferably reasoning about finite mathematical objects only—that no contradiction can be obtained in the formal system.

Hilbert's programme (cont.)

- Decidability: construction of an algorithm for deciding, in the formal system, the truth or falsity of any mathematical statement.
- Conservation: a proof that any result about "real objects" obtained using "ideal objects" (such as uncountable sets) can be restated without using ideal objects.

Hilbert's "axiom" (1900)

> *Is the axiom of solvability of every problem a peculiar characteristic of mathematical thought alone, or is it possibly a general law inherent in the nature of the mind, that all questions which it asks must be answerable? . . . This conviction of the solvability of every mathematical problem is a powerful incentive to the worker. We hear within us the perpetual call: There is the problem. Seek its solution. You can find it by pure reason, for in mathematics there is no* igorabimus.

By $N(P, v)$ we mean that the program P will *never* halt when begun with input v.

For any particular program P and input v, $N(P, v)$ is a perfectly definite statement which is either true (in case P will never halt in the described situation) or false (in case P will eventually halt).

When $N(P, v)$ is false, this fact can always be demonstrated by running P on v.

No amount of computation will suffice to demonstrate the fact that $N(P, v)$ is true. We may still be able to prove that a particular $N(P, v)$ is true by a logical analysis of P's behaviour, but, as we proved in the undecidability of the halting problem, no such method works correctly in all cases.

What is a *proof*?

Suppose that certain strings of symbols (possibly paragraphs of a natural language (English, for example)) have been singled out—typically with the help of axioms and rules of inference—as proofs of particular statements of the form $N(P, v)$.

Operationally, we assume that we have an algorithm called **syntactic test** that can test an alleged proof Π that $N(P, v)$ is true and determine whether Π is or is not actually such a proof.

There are two basic requirements which it is natural to demand of our supposed rules of proof:

- *Soundness:* If there is a proof Π that $N(P, v)$ is true, then P will in fact never halt when begun with v on its tape.
- *Completeness:* If P will never halt when begun with v on its tape, then there is a proof Π that $N(P, v)$ is true.

Theorem [Gödel's incompleteness theorem]

No rules of proof can be both sound and complete.

Obviously, soundness cannot be sacrificed.

If a given set of rules of proof is sound, then, according to Gödel's incompleteness theorem, there is some true statement $N(P, v)$ which has no proof Π according to the given rules of proof.

Such a true unprovable statement is called *undecidable* since it will surely not be disprovable.

Suppose we had found rules of proof which were both sound and complete.

Suppose that the "proofs" according to these rules were particular strings of symbols on some specific finite alphabet.

Let $\Pi_1, \Pi_2, \Pi_3, \ldots$ be the quasi-lexicographic computable enumeration of all finite strings on this alphabet. This sequence includes all possible proofs, as well as a lot of other things (including a high percentage of total nonsense). But, hidden among the nonsense, are all possible proofs.

Now we show how we can use our supposed rules of proof to solve the halting problem—an impossibility.

So, we are given a program P and an input v and wish to test whether or not P will eventually halt when begun on v.

We run in parallel two computations:

- the computation of P on v,
- we generate the sequence $\Pi_1, \Pi_2, \Pi_3, \ldots$ of possible proofs and, as each Π_i is generated, we use the **syntactic test** to determine whether or not Π_i is a proof of $N(P, v)$.

If P will eventually halt on v we will find out in finite time.

If P will never halt on v, since our rules of proof are assumed to be complete, there will be a proof Π_i of $N(P, v)$ which we will discover via the enumeration $\Pi_1, \Pi_2, \Pi_3, \ldots$ of possible proofs.

Having obtained this Π_i we will be sure (because of soundness) that P will indeed never halt.

Thus, we have described an algorithm which would solve the halting problem, a contradiction!

Gödel's theorem does not indicate any particular pair P, v for which we will never be able to convince ourselves that $N(P, v)$ is true!

It says that for any given sound rules of proof, there will be a pair P, v for which $N(P, v)$ is true, but not provable *using the given rules.*

There may well be, and in fact there always are, other sound rules which decide this "undecidable" statement. But these other rules will, in turn, have their own undecidabilities.

Incompleteness and its consequences showed that most of the goals of Hilbert's program are impossible to achieve, at least if interpreted in the "most obvious way".

However, much of it can be salvaged by changing slightly its goals:

- Although it is not possible to formalise all mathematics, it is possible to formalise essentially all the mathematics that "anyone uses". Zermelo-Fraenkel set theory, combined with first-order logic, gives a satisfactory and generally accepted formalism for essentially all current mathematics.

Hilbert's programme after incompleteness (cont.)

- Although it is not possible to prove completeness for systems at least as powerful as Peano arithmetic (if they have a computable set of axioms), it is possible to prove forms of completeness for many interesting systems.

  Gödel himself proved the completeness theorem for first-order logic before he proved the incompleteness theorems.

- The theory of algebraically closed fields of a given characteristic (e.g. the field of complex algebraic numbers) is complete.
- Although there is no algorithm for deciding the truth of statements in Peano arithmetic, Tarski's algorithm can decide the truth of any statement in analytic geometry (more precisely, the theory of real closed fields is decidable). Given the Cantor-Dedekind axiom, this algorithm can decide the truth of any statement in Euclidean geometry.

Objective vs. subjective mathematics

- Objective mathematics consists of the body of mathematical propositions, constructive or not, which hold true in an absolute sense. Peano Arithmetic or Zermelo-Fraenkel set theory are parts of it.
- Subjective mathematics consists of all mathematical truths *humanly demonstrable* (or *provable* or *knowable*), in a constructive manner or not.

Does objective mathematics coincide with subjective mathematics?

Gödel accepted Hilbert's rejection of the existence of absolutely unsolvable problems because otherwise,

> *it would mean that human reason is utterly irrational by asking questions it cannot answer, while asserting emphatically that only reason can answer them*

but found Turing's argument inconclusive:

> *Turing gives an argument which is supposed to show that mental procedures cannot go beyond mechanical procedures. However, this argument is inconclusive. What Turing disregards completely is the fact that mind, in its use, is not static, but constantly developing, i.e., we understand abstract terms more and more precisely as we go on using them . . . though at each stage the number and precision of the abstract terms at our disposal may be finite, both . . . may converge toward infinity . . .*

Gödel's answer (Gibbs lecture "Some Basic Theorems on the Foundations of Mathematics and their Implications", [2]) based on his incompleteness theorem is a disjunctive conclusion:

> *Either mathematics is incompletable in this sense, that its evident axioms can never be comprised in a finite rule, that is to say, the human mind (even within the realm of pure mathematics) infinitely surpasses the powers of any finite machine, or else there exist absolutely unsolvable diophantine problems of the type specified.*

Does objective mathematics coincide with subjective mathematics?

Martin-Löf's answer based on a *constructive* interpretation of the notions of "true", "false" and "can be known" [1]:

> *There are no propositions which can neither be known to be true nor be known to be false.*

For the non-constructive mathematician:

> *No propositions can be effectively produced (i.e. by an algorithm) of which it can be shown that they can neither be proved constructively nor disproved constructively. There may be absolutely unsolvable problems, but one cannot effectively produce one for which one can show that it is unsolvable.*

Guided by Post [2, p. 200],[3]

> *A fundamental problem is the question of the existence of absolutely undecidable propositions, that is, propositions which in some a priori fashion can be said to have a determined truth-value, and yet cannot be proved or disproved by any valid logic*

we will only require that the objective mathematics contains the subjective mathematics.

Furthermore, in contrast with Feferman [1], we will include in subjective mathematics all statements provable by any methods, axiomatic (dynamic, not only static), constructive, computational or by methods currently not yet discovered.

Constructive logical interpretations:

- The proposition A can be known to be true if we have a proof for A.
- The proposition $A \vee B$ can be known to be true if we have a proof for A or we have a proof for B.
- The proposition $A \wedge B$ can be known to be true if we have a proof for A and we have a proof for B.
- The proposition $A \to B$ can be known to be true if we have an algorithm which converts any proof for A into a proof for B.
- The proposition $\neg A$ can be known to be true if we have a proof for $A \to (0 = 1)$.

Martin-Löf's argument (cont.)

- The proposition A can be known to be false if we have a proof for $\neg A$.
- The proposition A cannot be known to be true if we have an algorithm which tests and rejects any given 'proof' which purports to demonstrate A.
- If the proposition A can be known to be true, then A is true.
- Martin-Löf's notions of can be known to be true/false are not related to any fixed formal system.

Fact 1. [Unknowability of truth entails knowability of falsity] *If the proposition A cannot be known to be true, then A can be known to be false.*

Proof: To prove that A can be known to be false we have to show that $\neg A = A \to (0 = 1)$ can be known to be true. To this aim we need an algorithm $\mathcal{B}$ to convert any proof of A into a proof of $(0 = 1)$. The algorithm $\mathcal{B}$ returns anything output by the algorithm $\mathcal{A}$ provided by the hypothesis, i.e. noting: vacuously, the implication holds.

Comment: The proof constructively produces positive information from negative information.

Fact 2. *If A can be known to be true and B can be known to be true, then $A \wedge B$ can be known to be true.*

Fact 3. [Absolute consistency] *The proposition $(0 = 1)$ cannot to be known to be true.*

Proof: The proposition $\neg(0 = 1)$ can be known to be true because $(0 = 1) \rightarrow (0 = 1)$ is provable using the identity algorithm, so $(0 = 1)$ can be known to be false, i.e. it is false. No proof can demonstrate $(0 = 1)$ because otherwise it would be true: the algorithm rejects any proof candidate.

Fact 4. [Law of contradiction] *One and the same proposition A cannot both be known to be true and be known to be false.*

Proof: By hypothesis we have a proof demonstrating A and a proof demonstrating $\neg A = A \rightarrow (0 = 1)$. Then we can demonstrate $(0 = 1)$, contradicting Fact 3.

Fact 5. [Law of excluded middle] *There is no proposition which can neither be known to be true nor be known to be false, i.e. there is no absolutely unprovable proposition.*

Proof: If A is a proposition which cannot be known to be true, then by Fact 1, A can be known to be false, a contradiction.


Solomon Feferman.
Are there absolutely unsolvable problems? Gödel's dichotomy.
Philosophia Mathematica, 14(2):134–152, 2006.


Kurt Gödel.
Some basic theorems on the foundations of mathematics and their implications.
In S. Feferman, J. W. Dawson, Jr., W. Goldfarb, C. Parsons, and R. M. Solovay, editors, *Collected Works. Unpublished Essays and Lectures. Volume III*, pages 304–323. Oxford University Press, 1995.


Per Martin-Löf.
Verification then and now.
In W. De Pauli-Schimanovich, E. Koehler, and F. Stadler, editors, *The Foundational Debate, Complexity and Constructivity in Mathematics and Physics*, pages 187–196. Kluwer Academic Publishers Dordrecht, 1995.


Emil L. Post.
Formal reductions of the general combinatorial decision problem.
American Journal of Mathematics, 65:197–215, 1943.


Alasdair Urquhart.
Emil Post.
In Dov M. Gabbay and John Woods, editors, *Logic from Russell to Church. Handbook of the History of Logic 5*, pages 617–666. Elsevier Science B.V., 2009.

Proof: 1. a fact or thing that shows or helps to show that something is true or exists; 2. a demonstration of the truth of something, "in proof of my statement"; 3. the process of testing whether something is true or good or valid, "put it to the proof". To prove: to give or be proof of; to establish the validity of; to be found to be, "it proved to be a good theory"; to test or stay out. To argue: 1. to express disagreement, to exchange angry words; 2. to give reasons for or against something, to debate; 3. to persuade by talking, "argued him into going"; 4. to indicate, "their style of living argues that they are well off". Argument: 1. a discussion involving disagreement, a quarrel; 2. a reason put forward; 3. a theme or chain of reasoning. (Oxford American Dictionary)

In all these statements, nothing is said about the means used "to show or help to show that something is true or exists", about the means used "in the process of testing whether something is true or good or valid".

In argumentation theory, various ways to argue are discussed, deductive reasoning being only one of them. We use suggestions, impressions, emotions, logic, gestures, mimicry, etc.

An *informal* (pen-on-paper) *proof* is a rigorous argument expressed in a mixture of natural language and formulae (for some mathematicians an equal mixture is the best proportion) that is intended to convince a knowledgeable mathematician of the truth of a statement, the theorem. Routine logical inferences are omitted. "Folklore" results are used without proof. Depending on the area, arguments may rely on intuition. Informal proofs are the standard of presentation of mathematics in textbooks, journals, classrooms, and conferences. They are the product of a social process.

A *formal proof* is written in a formal language consisting of certain strings of symbols from a fixed alphabet. Formal proofs are precisely specified without any ambiguity because all notions are explicitly defined, no steps (no matter how small) are omitted, no appeal to any kind of intuition is made. They satisfy Hilbert's criterion of mechanical testing:

> *The rules should be so clear, that if somebody gives you what they claim is a proof, there is a mechanical procedure that will check whether the proof is correct or not, whether it obeys the rules or not.*

By making sure that every step is correct, one can tell once and for all whether a proof is correct or not, i.e. whether a theorem has been proved.

Formal proof cannot be found in mathematical articles or books (except for a few simple examples).

However, most mathematicians believe that almost all "real" proofs, published in articles and books, can, with tedious work, be transformed into Hilbertian proofs. Why?

Because "real" proofs look *convincing* for the mathematical community. Going further, DeMillo, Lipton and Perlis argued that "real proofs" should be highly non-monolithic because they aim to be heard, read, assimilated, discussed, used and generalised by mathematicians—they are part of a social process.

Programming is the activity of solving problems with computers. It includes the following steps:

a) developing the *algorithm* to solve the problem,

b) writing the algorithm in a specific programming language (that is, coding the algorithm into a program),

c) assembling or compiling the program to turn it into machine language,

d) testing and debugging the program,

e) preparing the necessary documentation.

Ideally, at d) one should have written:

d) *proving the correctness of the algorithm*, testing and debugging the program.

We said, ideally, because correctness, although desired, is only practised in very few instances (for example, the programs involved in the proof of the Four-Color Theorem were not proved correct!)

It makes sense to prove the correctness of an algorithm, but *not*, the correctness of a program. Programs are analogues of mathematical models, they may be more or less adequate to code algorithms.

Adequacy is a property which depends on many factors, from pure formal/coding ones to physical and engineering ones.

One can even argue that a "correctness proof" for a program, if one could imagine such a thing, adds very little to the confidence in the program. In Knuth's words:

> *Beware of bugs in the above code: I have only proved it correct, not tried it.*

The role of proof in mathematical modelling is very small: *adequacy is the main issue!* Here is an illustration from Jack Schwartz:

> *. . . it may come as a shock to the mathematician to learn that the Schrödinger equation for the hydrogen atom . . . is not a literally correct description of this atom, but only an approximation to a somewhat more correct equation taking account of spin, magnetic dipole, and relativistic effects; that this corrected equation is itself only an ill-understood approximation to an infinite set of quantum field-theoretical equations; and finally that the quantum filed theory besides diverging, neglects a myriad of strange-particle interactions whose strength and form are largely unknown. . . .*

The modelling component of mathematics appears not only in applications, but also in the way mathematics develops new concepts.

Many important notions in mathematics reached an accepted definition only after a long process of modelling, from an intuitive, pre-mathematical notion to a more precisely defined, formal one. In the end, the accepted definition is adopted as a "thesis" claiming its adequacy.

For example, "Weierstrass' thesis" is the statement that the intuitive notion of continuity is extensionally equivalent to the notions yielded by the now standard definitions of continuous function.

Other examples include:

- "The function thesis": identification of a function with a set of ordered pairs,
- "Tarski's thesis": identification of Tarski's definition of truth in a formalised language with the intuitive notion of truth,
- "Church-Turing thesis".

None of these "theses" can be *proved*, but various analyses can conclude their degrees of plausibility/adequacy/applicability. Mathematics in both its practice and development is an "open-texture" .

There are many "new" types of proofs, probabilistic, experimental or hybrid proofs (computation plus theoretical arguments).

Zeilberger has argued in favour of the transition from rigorous proofs to an "age of *semi-rigorous* mathematics, in which identities (and perhaps other kinds of theorems) will carry price tags" measured in computer and human resources necessary to prove them with a certain degree of confidence.

> *The real work of us mathematicians, from now until, roughly, fifty years from now, when computers won't need us anymore, is to make the transition from human-centric math to machine-centric math as smooth and efficient as possible.*

No theorem is validated before it is "communicated" to the mathematical community (orally and, eventually, in writing). Manin:

> *Proof is not just an argument convincing an imaginary opponent. Not at all. Proof is the way we communicate mathematical truth.*

However, as Rota pointed out:

> *One must guard, however, against confusing the presentation of mathematics with the content of mathematics.*

Proofs have to be written on paper, which means proofs are *physical*. From this perspective, proofs depend upon the physical universe (Calude and Chaitin).

Standards of rigour have changed throughout the history of mathematics and not necessarily from less rigour to more rigour.

B. Russell:

> *I wanted certainty in the kind of way in which people want religious faith.*

J.-P. Serre was quoted saying that mathematics is the only producer of "totally reliable and verifiable" truths.

D. Knuth:

> *. . . programming demands a significantly higher standard of accuracy. Things don't simply have to make sense to another human being, they must make sense to a computer.*

W. P. Thurston:

> *The standard of correctness and completeness necessary to get a program to work at all is a couple of orders of magnitude higher than the mathematical community's standard of valid proofs.*
> *When one considers how hard it is to write a computer program even approaching the intellectual scope of a good mathematical paper, and how much greater time and effort have to be put into it to make it "almost" formally correct, it is preposterous to claim that mathematics as we practice it is anywhere near formally correct.*

A conceptual period reaches its maturity under the form of an operational one, which, in its turn, is looking for a new level of conceptual attitude.

The whole treatise of Bourbaki is a conceptual reaction to an operational approach. Dirichlet's slogan asking to replace calculations with ideas should be supplemented with another, complementary slogan, requiring to detect an algorithmic level of concepts.

Can we expect a similar alternation of attitudes in respect to programming? Perhaps it is too early to answer, taking into account that the whole field is still too young. The question is not only academical as the project Flyspeck (to produce a formal proof of the Kepler Conjecture completed in 2014) reminds us.

In theory, each informal proof can be converted into a formal proof. However, this is rarely, almost never, done in practice.

A proof assistant (interactive theorem prover) is a software tool to assist with the development of formal proofs by man-machine collaboration.

Proof-assistants can be used not only to check the validity of a formalised proof of a known mathematical result, but also to interactively help to "prove" new theorems.

An impressive record of deep mathematical theorems formally proved:

- Gödel Incompleteness Theorem (1986),
- the Fundamental Theorem of Calculus (1996),
- the Fundamental Theorem of Algebra (2000),
- the Four Colour Theorem (2004),
- Jordan's Curve Theorem (2005),
- the Prime Number Theorem (2008).

The December 2008 issue of the *Notices of AMS* includes four papers on formal proof.

Produced by INRIA, free distributed under the GNU Lesser General Public Licence, http://coq.inria.fr/, Coq is a formal proof management system providing a formal language to write mathematical definitions, executable algorithms and theorems together with an environment for semi-interactive development of machine-checked proofs.

Typical applications include the formalisation of programming languages semantics, formalisation of mathematics (e.g. Four Colour Theorem) and teaching.

Isabelle is a generic proof assistant, free distributed under the BSD license, http://www.cl.cam.ac.uk/research/hvg/Isabelle/, allowing mathematical formulas to be expressed in a formal language and provides tools for proving those formulas in a logical calculus.

Isabelle is developed at the University of Cambridge, Technische Universität München and Université Paris-Sud.

The first formal proofs in algorithmic information theory have been developed in Isabelle at UoA (C. Calude and N. Hay).

George Box: All models are wrong, but some are useful.

Models have been able to consistently, if imperfectly, explain the world around us.

Is there any choice?

Sixty years ago, digital computers made information readable.

Twenty years ago, the Internet made it reachable.

Ten years ago, the first search engine crawlers made it a single database.

Kilobytes are stored on floppy disks, megabytes are stored on hard disks, terabytes are stored in disk arrays, and petabytes are stored in the cloud. We leave in the *Petabyte Age*.

Every day, we create 2.5 quintillion bytes of data—so much that 90% of the data in the world today has been created in the last two years alone. This data comes from everywhere: sensors used to gather climate information, posts to social media sites, digital pictures and videos, purchase transaction records, and cell phone GPS signals to name a few. This data is big data. IBM: What is big data?

We don't know why this page is better than that one. If the statistics of incoming links say it is, *that's good enough*. *No semantic or causal analysis is required*.

Operationalising this philosophy, Google

- can match ads to content without any knowledge or assumptions about the ads or the content;
- can translate languages without actually "knowing" them; it can translate Maori into Farsi as easily as it can translate French into English provided it has equal corpus data (see WIKI-LINKS).

According to C. ANDERSON, *Wired Magazine*, Google's research director Peter Norvig offered an update to George Box's dictum:

> *All models are wrong, and increasingly you can succeed without them.*

Unfortunately, Norvig denies:

> **That's a silly statement, I didn't say it, and I disagree with it.**

The big target here is science. The scientific method is built around testable hypotheses. The models are then tested, and experiments confirm or falsify theoretical models of how the world works. This is the way science has worked for hundreds of years.

THE END OF THEORY:
THE DATA DELUGE MAKES THE SCIENTIFIC METHOD OBSOLETE.

Welcome to data science! Long live the dead science!

Operationalising the idea

THE END OF THEORY:
THE DATA DELUGE MAKES THE SCIENTIFIC METHOD OBSOLETE.

In short, the more we learn about [[biology]], the further we find ourselves from a model that can explain it.

There is now a better way. Petabytes allow us to say: "Correlation is enough." We can stop looking for models. We can analyze the data without hypotheses about what it might show. We can throw the numbers into the biggest computing clusters the world has ever seen and let statistical algorithms find patterns where science cannot.

NSF : *Computational and Data-Enabled Science and Engineering (CDS & E) is a new program. CDS & E is now clearly recognizable as a distinct intellectual and technological discipline lying at the intersection of applied mathematics, statistics, computer science, core science and engineering disciplines.. . . We regard CDS & E as explicitly recognizing the importance of data-enabled, data-intensive, and data centric science. CDS & E broadly interpreted now affects virtually every area of science and technology, revolutionizing the way science and engineering are done. Theory and experimentation have for centuries been regarded as two fundamental pillars of science. It is now widely recognized that computational and data-enabled science forms a critical third pillar.*

The "wave of the future"—as it was called—disposes of experiment and theory in science. It affects everything from astronomy to zoology, from medical sciences to social sciences.

History reminds us of similar exaggerations (recall chaos theory or catastrophe theory in mathematics): the "wave of the future" often washes away a number of worthy things and leaves a number of questionable items littering the shore.

In June 2012, Google demonstrated the power of "data-oriented deep learning" with one of the largest neural networks containing more than a billion connections.

A Stanford University–Google team (lead by Andrew Ng and Jeff Dean) showed the system images from 10 million randomly selected YouTube videos. One simulated neurone in the model fixated on images of cats. Others focused on human faces, yellow flowers, and other discrete objects.

The model identified reasonable well these discrete objects even though no humans had ever defined or labeled them.

Data science: translating is't easy...

How computers translate human language

Does grammar matter?

Imagine there are two papers somewhere in the literature, one of which says that *A implies B*, and another that says *B implies C*. With the incredible growth of the scientific literature, it is likely that these two papers remain unrelated.

A program can find a way to stitch these two papers together, showing that *A implies C*, potentially an important discovery.

Is there a price in this finding? Cornell University program EUREQA is a free tool for detecting equations and hidden mathematical relationships in data with the goal "to identify the simplest mathematical formulas which could describe the underlying mechanisms that produced the data".

A theorem may be proven in this way, but no one person actually may understand the proof, though there may be reasons to believe it is correct.

So what does this all mean for the future of truth?

Is it possible for something to be true but not understandable? Is this bad?

Believing without finding reasons is controversial. However, if these findings motivate the search for more elegantly constructed, human-understandable, versions of these proofs, then they are good.

Data are assumed to accurately reflect the "real world"; however, significant gaps, with little or no signal coming from particular parts may exist.

Boston has a problem with potholes, patching approximately 20,000 every year. STREETBUMP smartphone app passively detects and instantly reports potholes to the City. *A clever approach which has a signal problem.* People in lower income groups are less likely to have smartphones, or to use StreetBump and this is particularly true of older residents, where smartphone penetration is as low as 16%.

Smartphone data sets miss inputs from significant parts of the population.

Ramsey theory, named after the British mathematician, logician and philosopher Frank P. Ramsey, is a branch of mathematics that studies the conditions under which order must appear.

Problems in Ramsey theory typically ask questions of the form:

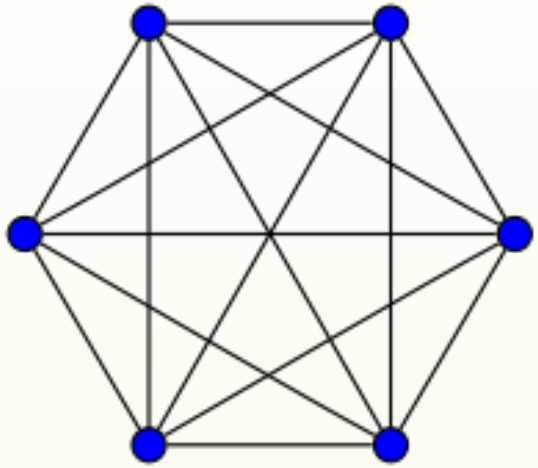
> *How many elements of some structure must there be to guarantee that a particular property will hold?*

Suppose a party has six people. Consider any two of them.

They might be meeting for the first time—in which case we will call them *mutual strangers*; or they might have met before—in which case we will call them *mutual acquaintances*.

Theorem (Paul Erdös, Alfréd Rényi, Vera T. Sós): *In any party of six people either*

- *at least three of them are (pairwise) mutual strangers or*
- *at least three of them are (pairwise) mutual acquaintances.*



Party graph

This is the complete graph with six vertices in which every pair of vertices is joined by an edge. Every colouring of edges with red and blue, cannot avoid having either a red triangle or a blue triangle.

Proof. Choose any one vertex; call it P. There are five edges leaving P. They are each coloured red or blue. The *pigeonhole principle* says that at least three of them must be of the same colour.

Let A, B, C be the other ends of these three edges, all of the same colour, say blue. If any one of AB, BC, CA is blue, then that edge together with the two edges from P to the edge's endpoints forms a blue triangle. If none of AB, BC, CA is blue, then all three edges are red and we have a red triangle, namely, ABC.

When a set system is "quite big"?

Given a set X, a collection of subsets $S \subset 2^X$ is called *partition regular* if every set $A \in S$ has the property that, no matter how A is partitioned into finitely many subsets $A = C_1 \cup C_2 \cup \cdots \cup C_n$, at least one of the subsets C_i must belong to the collection S.

The infinite pigeonhole principle. Partition regularity asserts that every finite partition of an infinite set contains an infinite set. Proof: take S the collection of all infinite subsets of the infinite set.

Ramsey theory proves that

> *Complete disorder is an impossibility. Every large set of numbers, points or objects necessarily contains a highly regular pattern.*

R. Graham, J. H. Spencer. Ramsey Theory, *Scientific American* 262 no. 7 (1990), 112–117.

- How to distinguish correlation from causation?
- How to distinguish content-correlations from Ramsey-type correlations?

A growing list of over 24,000 spurious correlations in real databases:

- Total revenue generated by arcades (US) correlates with Computer science doctorates awarded (US). Correlation: 0.985065
- Per capita consumption of mozzarella cheese (US) correlates with Civil engineering doctorates awarded (US). Correlation: 0.958648
- Precipitation in Yakima County, WA correlates with Apple iPhone sales. Correlation: 0.993657
- Divorce rate in Maine correlates with Per capita consumption of margarine (US). Correlation: 0.992558
- People who drowned after falling out of a fishing boat correlates with Marriage rate in Kentucky. Correlation: 0.952407

Using classical results from ergodic theory, Ramsey theory and algorithmic information theory, the paper

C. S. Calude, G. Longo. The deluge of spurious correlations in big data, *Foundations of Science*, 2016, DOI 10.1007/s10699-016-9489-4.

proves that very large databases contain arbitrary correlations due to the size, not the nature, of data. They can be found in "randomly" generated, large enough databases, which implies that a) they are spurious, and b) most correlations are spurious.

Too much information tends to behave like very little information. The scientific method can be enriched by computer mining in immense databases, but not replaced by it.

Calling Bullshit: Data Reasoning in a Digital World

A cooking model would include all available food options, nutritional values and costs, a data base of a family culinary tastes (preferences and aversions).

As data can (and will) vary in time, a good solution is to design a "trainable model" by entering data every day (ingredients, meals cooked, family members reactions). *Parameters and constraints* will be introduced.

For example, seasonal limits on fruits and vegetables, restrictions on sugar, with care not to face an open rebellion. Rules can be used to mitigate different preferences, like "one likes meat, another one wants bread and pasta, one cannot drink milk".

Conditions changes, so the model must change.

In time this formal model will be trained to produce an acceptable cooking for the whole family.

Such a model is an automated cooking version of its author which others can implement and use too. This adds power to the model.

There will be mistakes, as any model is a *simplification*. Choices are used to create a model. Occasionally, the model can do its job in a cluelessly manner, with enormously *blind spots*—those parts which by design are left out.

For example, avionics software models the winds, the plane's speed, the landing strip below, but ignores deliberately streets, tunnels, buildings, or people.

Some choices appear natural, like the ones made in avionics software models, but others are far more problematic.

A "value-added" model used in Washington D.C. schools to evaluate teachers is based on students' test scores, but ignores the degree of engagement of teachers with students and classroom management work on specific skills.

It's simple, but it sacrifices accuracy and insight for efficiency. It is cost-effective for administrators who can tolerate the risk of "misreading" of some teachers (who are categorised as "under-performers", so fired).

Models aspire to a reputation of impartiality, but in fact they reflect goals and ideology. They are opinions embedded in mathematics.

Models are not true or false, they a adequate or not. Judging them is not a matter of mathematics, but of opinion.

According to

Cathy O'Neil. *Weapons of Mass Destruction: How Big Data Increases Inequality and Threatens Democracy*, Crown, New York, 2016

models can be analysed according to

- opacity,
- scale,
- damage.

Opaque and invisible models are the rule—it's "intellectual property" that must be defended by all means (see Google, Amazon, Facebook). Another important question is whether the model is based on proxies or not.

Scale is paramount because models can leap from one field to another: from epidemiology to box office predictions, from viruses to computer viruses (a model for identifying AIDS viruses is successfully used for spam filters). The world of finance is another example.

Is the model fair? Does the model damage or destroy lives?

In 2017 AI has promised revolutions in health and driving.

- According to Jerome Pesenti, chief executive of healthcare firm Benevolent AI, health is on the verge of a revolution because

  > *Coming up with new ideas is tedious and serendipitous—a machine comes up with a list much faster.*

  So far we have seen little action.
- 'Autonomous car' is the buzz word for many car makers, Tesla, but not only. Again, the promises are very high. We have seen significant progress, but also challenges.

The rapid progress of AI is not matched by winning public trust in the technology.

Already a few years ago Bill Gates, Stephen Hawking, Elon Musk and Steve Wozniak have warned that AI could destroy humanity.

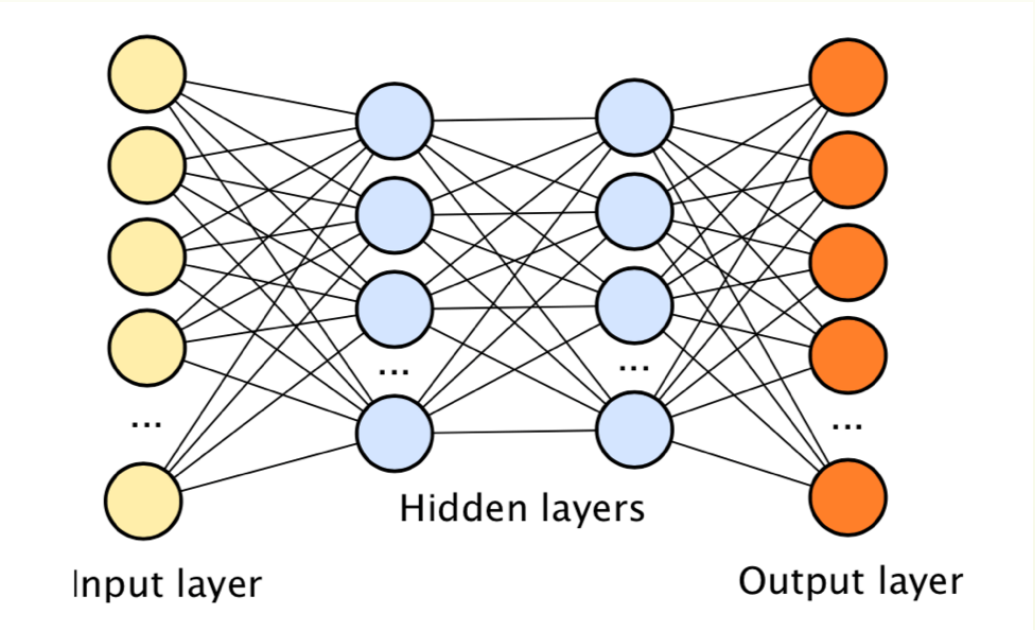
The potential for bias in algorithms which invisibly collect information and deliver all sorts of services is not just a theoretical concern. Fake news is an example: bots are used to spread nonsense content that can (maybe already did) undermine democracy.

Hoaxes are presented as fact, conspiracy theories are offered as truth. However, according to Jimmy Wales, co-founder of Wikipedia, one of the most-visited sites online, while not immune, has had almost no problems with this phenomenon.

> *Simply because our community is quite you know, it's their hobby to debate about the quality of sources, and it's very difficult to fool the Wikipedia community with this.*

Deep (hierarchical) learning is a type of machine learning based on learning data representations, as opposed to task-specific algorithms. Learning can be supervised, semi-supervised or unsupervised.

Deep learning algorithms use a cascade of successive multiple layers of nonlinear processing units for feature extraction and transformation. The output from the previous layer is an input for the next one.



Deep learning algorithms

1. learn is supervised (e.g. classification) and/or unsupervised (e.g. pattern analysis) manners;
2. learn multiple levels of representations that correspond to different levels of abstraction (a hierarchy of concepts).

Deep learning algorithms have be trained with large sets of labelled data. For example, you have to give the algorithm thousands of pictures of cats before it can start classifying new cat pictures with relative accuracy. The larger the training data set, the better the performance of the algorithm.

Layers include hidden layers of an artificial neural network, sets of propositional formulas, latent variables organised layer-wise (as in deep generative models Deep Belief Networks and Deep Boltzmann Machines).

Deep learning has been successfully used in automatic speech recognition, image recognition, natural language processing, games (specifically Go, AlphaGo).

Shane Legg, one of the co-founders of the DeepMind got his MSc with the Thesis Solomonoff Induction, supervisor C. Calude, at the University of Auckland in 1997. Here is DeepMind explanation how does AI works. Shane will give a public talk on Monday 7 May 2018, 2.00-3.00 pm, in the Conference Centre, 423–342.

"In a world with infinite data, and infinite computational resources, there might be little need for any other technique." (G. Marcus).

And here lies the problem: *we don't live in such a world.* The sample training space is incomplete, hence the algorithm has to generalise and/or interpolate between its previous samples in order to classify data it has never seen before.

So what happens when the deep learning algorithm doesn't have enough quality training data? It can fail like thinking that a riffle is a helicopter see http://tinyurl.com/yak5c7ls.

A main criticism of deep learning concerns the lack of theory underlying their methods.

The well-understood gradient descent is used in most common deep architectures. However, other algorithms like contrastive divergence, are less researched. Do they converge? If so, how fast? What is it approximating?

Deep learning methods are often looked at as a black box, for which most confirmations are done empirically, rather than theoretically. These algorithms sift through millions of data points to find patterns and correlations that often go unnoticed to human experts.

If the algorithm uses multiple layers and a huge data base, then there is no way to understand why it has chosen a certain solution – decision they make often surprise even the scientists who created them – or even whether the solution is correct or not.

These algorithms often inherit the biases of their training data, a problem that is very hard to debug.

Given that some deep learning architectures display problematic behaviours, their use, say in medical sciences, could be rather troublesome.

An autonomous (driverless) car is a vehicle capable of sensing its environment and navigating without human input.

Autonomous cars use various techniques to detect their surroundings, such as radar, computer vision, laser light, GPS.

Experiments on automating driving go back till since at least the 1920s. The first autonomous prototype cars appeared in the 1980s at Carnegie Mellon University and Mercedes-Benz and Bundeswehr University Munich in 1987. Since then, numerous companies and research organisations have developed prototypes. A history of autonomous cars.

In 2017 Audi announced that the latest A8, using its "Audi AI", will be autonomous at up to speeds of 60 km/h.

AI has surpassed humans in image and speech recognition but it is less good in predicting what comes next in a fast changing environment.

For example, humans are good at hearing a siren behind them and pulling over to let an ambulance pass. Motorists can negotiate through subtle gestures and body language which is difficult to model and replicate. Autonomous cars struggle in both cases.

In California only there have been over 30 accidents involving self-driving vehicles since 2014 and the number is increasing.

In November 2017 a self-driving shuttle bus in Las Vegas was involved in a crash on its first day of service.

I December 2017 a Chevrolet Bolt that was driving autonomously collided with a motorcycle as the car was changing lanes.

In January 2018 a Tesla Model S and a Chevrolet Bolt have been involved in separate accidents. In both cases, the vehicles used their respective autonomous driving systems.

Who is the responsible party? The driver or the automakers that designed the autonomous driving technology?

No-win choices in driving are rare, but not impossible. Who decides who dies in a crash? Where a moral snap judgment must be made? Legislators, government, city councils, programmers, philosophers. . . ?

Who bears responsibility? There are many ways you could answer ethical questions about autonomous cars—more generally, autonomous devices—leading to different outcomes.

Here are some ethical questions:

1. Which risks are worth taking?
2. Is the car making a choice or are we?
3. Is there a moral code we can all agree on?
4. Or should we choose our own car's moral code?
5. Can we ever learn to trust our self-driving cars?

Here is an MIT moral machine to test your answers.

Trust can be attributed to relationships between people. Can we extrapolate this attribute to the world of AI?

An IBM team proposed four fundamental pillars to *trusted AI* https://arxiv.org/pdf/1808.07261.pdf, Feb. 2019:

1. Fairness: AI systems should use training data and models that are free of bias, to avoid unfair treatment of certain groups.
2. Robustness: AI systems should be safe and secure, not vulnerable to tampering or compromising the data they are trained on.
3. Explainability: AI systems should provide decisions or suggestions that can be understood by their users and developers.
4. Lineage: AI systems should include details of their development, deployment, and maintenance so they can be audited throughout their lifecycle.