

BRK-016 · Formal proof encounters internal limits

BRK · Conceptual Breaks

Juan Segurola · All Rights Reserved

A rule-governed method can be precise without being able to settle every question expressible within it. Twentieth-century logic made this limitation a mathematical object. Instead of merely failing to find a proof or an algorithm, investigators could prove that a specified class of methods could not accomplish a universal task.

The resulting change did not abolish proof. It distinguished several achievements that can look interchangeable from a distance: checking a proposed derivation, finding a derivation, deciding every statement and establishing a system's consistency. Once these tasks were separated, their different limits could be demonstrated.

1. Studying proofs as finite objects

Hilbert's program sought to formalize mathematical reasoning and justify its consistency using restricted, finitary methods. Its ambitions were more specific than the popular image of a machine automatically discovering every truth. The formalized proofs themselves could be studied as finite sequences, even when the mathematics they expressed concerned infinite structures. [1]

This shift made a proof do double duty. It was an argument within mathematics and an object for another mathematical investigation. One could ask whether a sequence obeyed the rules, whether a contradiction could ever be derived and which assumptions a proposed consistency argument required.

The program did not treat arbitrary symbol manipulation as sufficient for meaningful science. It sought a justification of powerful mathematical methods. Its difficulties therefore concern the reach of particular formal and justificatory schemes, rather than the claim that all mathematical practice is nothing but syntax.

2. Incompleteness is relative to a system

Gödel's 1931 work established limitations for formal systems capable of representing enough arithmetic. In the modern form incorporating Rosser's strengthening, a consistent, effectively axiomatized theory extending suitable elementary arithmetic cannot decide every sentence in its language: some sentence is neither provable nor refutable within it. [2,3]

Each condition matters. A weak or restricted theory need not be incomplete in this way. A collection containing every true arithmetic statement is not an effectively axiomatized escape from the result. And an inconsistent system is not a successful complete foundation merely because classical inference permits it to derive anything.

The word "within" matters as much as the word "cannot." A sentence unprovable in one theory may be provable in a stronger theory. Adding it as an axiom changes the theory. If the new theory still satisfies the relevant conditions, incompleteness applies again. The theorem does not identify one sentence that no conceivable form of reasoning could ever establish.

3. A system talks about its own derivations

The technical route uses arithmetic to encode expressions and proofs. Relations between finite symbol sequences can thereby be represented as relations between numbers. Diagonal construction

then produces a sentence whose arithmetic content corresponds to a claim about its own provability in the selected system. [2]

This is not simply the everyday sentence “this sentence is false” dressed in mathematical notation. Provability is a specified relation involving coded derivations and rules. The construction works because those relations can be represented with sufficient precision inside arithmetic. The conditions of the proof prevent a verbal paradox from standing in for an argument.

Gödel’s original treatment used a stronger hypothesis than bare consistency for one side of the first result. Later formulations can weaken that requirement. A responsible historical account should distinguish the original proof from the streamlined theorem commonly taught today, rather than attribute every later refinement to 1931. [3]

4. Consistency cannot simply certify itself

The second incompleteness theorem places a further limit on sufficiently strong consistent theories: under the usual formalization of proof and consistency, such a theory cannot prove its own consistency. The qualification about formalization is essential; writing any convenient sentence and calling it “consistency” does not reproduce the theorem’s target. [3]

This does not mean that a mathematician cannot give a consistency proof for a theory. A stronger framework can establish the consistency of a weaker one, and relative consistency results remain valuable. What is blocked is a particular kind of self-certification from within a system satisfying the theorem’s assumptions.

Nor does the result render individual proofs unreliable. If a finite derivation correctly follows accepted rules from accepted axioms,

the theorem does not insert an error into it. It constrains what the total formal apparatus can establish about itself and about all sentences in its language.

5. Computation acquires an analogous limit

Church and Turing approached effective calculation through different formal analyses in the 1930s. Turing's 1936 paper described computation in terms of a machine following finite instructions and used that analysis to establish an undecidability result. The contribution was not merely a proposed piece of hardware. It made a general limitation of mechanical procedures mathematically accessible. [4,5]

A modern halting argument illustrates the structure. Suppose a procedure H always correctly decides whether any program halts on any input. Construct a program D that, on input a program description p , asks H what p does on input p . If H predicts halting, D runs forever; if H predicts non-halting, D halts.

Now give D its own description. Either answer from H makes D behave contrary to that answer. Therefore the supposed universal, always-correct decider cannot exist. This is an explanatory modern formulation, not the wording or exact target of Turing's original proof. It concerns an unrestricted class of programs; many particular programs and restricted classes can still be analyzed successfully.

The contrast with checking a proof is instructive. A valid proof, when supplied, can have a finite mechanically checkable structure even when no algorithm decides every possible mathematical question. Verification of a presented object and guaranteed decision of an arbitrary question are different tasks.

6. What the limits made possible

The Church–Turing thesis connects an informal idea of effective calculation with precise models of computation. Equivalence results between formal models are mathematical theorems; the identification with the informal notion is a thesis supported by analysis, not another instance of the same kind of formal proof. [5]

These results do not establish that human minds necessarily transcend every computational account. Such a conclusion requires additional assumptions about human reasoning and its consistency. They also do not show that truth is arbitrary or that nothing can be known. Their force depends on the precision of the propositions they actually establish.

The conceptual break was that limits became demonstrable from inside mathematical inquiry. Failure to finish a universal program could be explained by a theorem rather than by insufficient persistence. That redirected work toward stronger and weaker systems, restricted decision procedures and explicit assumptions, while leaving an enormous domain of rigorous proof intact.

Related reading

BRK Article 017, “Making sensation measurable: psychophysics and mental chronometry,” examines a different boundary between an observable operation and the stronger inference one might be tempted to draw from it.

References

- [1] Zach, R. “Hilbert’s Program.” *Stanford Encyclopedia of Philosophy*. <https://plato.stanford.edu/entries/hilbert-program/>
- [2] Gödel, K. “Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I.” *Monatshefte für Mathematik und Physik* 38, 173–198 (1931). <https://doi.org/10.1007/BF01700692>

- [3] Raatikainen, P. "Gödel's Incompleteness Theorems." *Stanford Encyclopedia of Philosophy*. <https://plato.stanford.edu/entries/goedel-incompleteness/>
- [4] Turing, A. M. "On Computable Numbers, with an Application to the Entscheidungsproblem." *Proceedings of the London Mathematical Society*, series 2, 42, 230–265 (1936–1937). <https://doi.org/10.1112/plms/s2-42.1.230>
- [5] Copeland, B. J. "The Church-Turing Thesis." *Stanford Encyclopedia of Philosophy*. <https://plato.stanford.edu/entries/church-turing/>