Can AI be creative? Gödel's incompleteness theorem, Art, and AlphaGo

February 4, 2021

PHIL3750: Philosophy of Artificial Intelligence

Michael Barkasi

barkasi@yorku.ca

York University, Toronto

1. The early origins: Lovelace, machine learning, and theorem proving
2. Gödel and his theorem
3. Two replies to the Gödelian argument
4. What is creativity?
5. Modern examples: Art, AlphaGo, and others

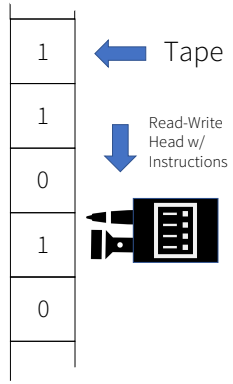
2. Gödel and his theorem

Kurt Gödel was one of the most influential logicians of the twentieth century. Along with Turing and others, he shaped early discussions of computability and AI. Specifically, he proved his famous incompleteness theorems, along with important results on the continuum hypothesis, intuitionistic logic, and even found one of the few exact solutions to Einstein's equations of general relativity.

Gödel, like Turing, worked in the 1930s to understand formal computation. While Turing is best known for giving a definition of which operations are computable by a machine, Gödel is best known for finding limits of what could be computed by such operations.

Gödel proved a number of results, but here is the most famous, what people often refer to as Gödel's incompleteness theorem. (There are several incompleteness results proved by Gödel.)

Recall that purely formal, or mechanical, operations are performed over strings of syntactically defined symbols. For example, Turing machines manipulate a tape divided into individual cells, each cell with a mark (e.g., a number) printed in it.



Among other tasks, such a system of formal operations, like a Turing machine, can be used to derive theorems of logic. That is, it can do automated theorem proving.

The symbols of logic include things like:

- NOT, AND, OR, IFF, and IF-THEN, along with
- symbols for predicates like CAT, HOUSE, or RED,
- variables like X, Y, and Z,
- names like JOHN, JAN, and JUAN, and
- quantifiers like ALL and THERE-EXISTS.

These symbols are put into strings (sentences) like IF-THEN(HOUSE(X),BUILDING(X)).

A sentence of logic is a theorem if and only if it can be gotten (i.e., derived, or proved) by applying the formal rules of the system in some order.

Sample formal rules:
IF-THEN(ϕ,ψ), $\phi \vdash \psi$
AND(ϕ,ψ) $\vdash \phi, \psi$
NOT(NOT(ϕ)) $\vdash$ NOT(ϕ)

Now consider what's normally called a Gödel sentence, G. G is some string of symbols, in the language used by our Turing machine, which we'd intuitively describe as asserting its own underivability. Roughly, G says something like "This sentence is unprovable in the formal system expressing me."

Now, there are many possible sets of symbols, many possible ways to put them together into sentences, and many possible rules of inference. A formal system is one such set of symbols, sentences, and inference rules with which you can do some logic.

Not every formal system has a Gödel sentence G. To have a Gödel sentence, a system must be able to "talk about itself". For each possible sentence ϕ in the system, the system needs another symbol Φ that's a name for that sentence. The system also needs a predicate THEOREM that applies to all and only those sentences which can be derived in the system.

The system will need to operate as expected with these additional symbols. For example, if a sentence ϕ (named by Φ) is a theorem in the system, then the system should also be able to derive THEOREM(Φ), and vice versa. We'll say that such a system is self-referentially representable; i.e., it can talk about (represent) itself.

One of Gödel's insights was figuring out how build a self-referentially representable formal system. He did this through what we now call Gödel numbering, but we won't go into the details.

So, a self-referentially representable formal system will have some symbol which names the Gödel sentence G. For example, maybe the name of G within the symbols of the system is GÖDEL.

How do we know that a self-referentially representable formal system has a Gödel sentence? This is another result Gödel proved. For our purposes, it suffices to note that in any self-referentially representable system with a theorem predicate, there must exist a sentence ϕ with name Φ such that the sentence IFF(ϕ,NOT(THEOREM(Φ))) is derivable. This sentence ϕ is, of course, the Gödel sentence G, and its name Φ is GÖDEL.

What Gödel proved is that this sentence G within a self-referentially representational system cannot be derived within the system. This result is normally what's called Gödel's incompleteness theorem.

All the hard work goes into showing how to build self-referentially representable systems with Gödel numbering and showing that Gödel sentences exist within them. Once you show that much, proving that a system can't derive its own Gödel sentence is easy.

Proof: Assume the system can derive G. As already noted, the system can also derive IFF(G, NOT(THEOREM(GÖDEL))). A basic inference rule of this system will be IFF(ϕ,ψ), $\phi \vdash \psi$. Hence, by applying this rule the system can derive NOT(THEOREM(GÖDEL)). But since G can be derived (so we're assuming) and the system is self-referentially representable, THEOREM(GÖDEL) can be derived as well. So, if the system can derive G, it can also derive a contradiction. So, if a self-referentially representable system is consistent, it can't derive G. □

Now notice something curious about the Gödel sentence G: by proving Gödel's incompleteness theorem, we've proved that G is true! G asserts that it itself is underivable in its formal system, and by proving the incompleteness theorem we've proved that it is, in fact, underivable. Hence, G is true, and we know it.
*Well, there's a catch…

If we think about the formal system, of which G is a part, as an actual physical computer, like a Turing machine or my MacBook, what Gödel's incompleteness theorem says is that this physical machine cannot, following its own rules for manipulating its own language, ever output the sentence G. G is a true sentence of the system's language, but not one the system can figure out is true!

In his full proof, Gödel shows that any formal system expressive enough to capture basic arithmetic is self-referentially representable, and hence has a Gödel sentence G; hence, his incompleteness theorem is often put as follows: In any consistent formal system strong enough to capture basic arithmetic, there will be true sentences the system can't prove. In other words, there is no single, consistent formal system which can prove all the truths about mathematics (even simple math about arithmetic)!

Gödel published his famous results in a 1931 paper titled "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme". Translated: "On Formally Undecidable Propositions of Principia Mathematica and Related Systems"

He initially presented the results during a talk he gave in Königsburg in September 1930. He was only 24 years old.

While Gödel executed on them and his Gödel numbering technique was key, the ideas for this proof were in the air at the time and arose organically over a few months of discussion between Gödel and other logicians, especially John von Neumann (who sketched his own incompleteness proofs).

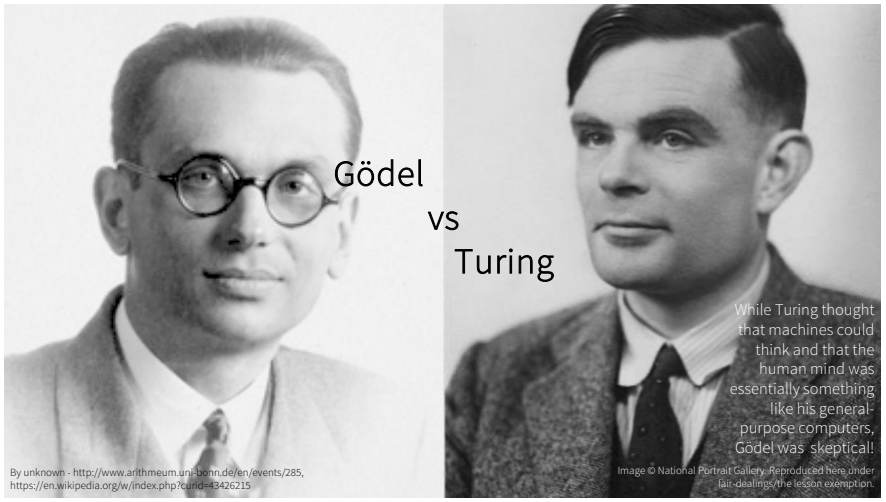
It's not hard to draw implications from Gödel's theorem for AI. Namely, G looks to be a sentence that we, human minds, have proved to be true, but which the formal system expressing that sentence cannot itself figure out is true. For example, a Turing machine working in a self-referentially representable system could not print G on its tape! Many people (including famous intellectuals, like Roger Penrose) are tempted to jump from this result to the conclusion that human minds can't be formal machines.

In fact, Gödel himself was tempted to interpret his own results as showing that human minds aren't formal machines (Raatikainen 2020)! To his credit, Gödel was much more cautious than most others, and didn't think the incompleteness theorem on its own showed this.

Raatikainen (2020) "Gödel's Incompleteness Theorems", The Stanford Encyclopedia of Philosophy (Winter 2020 Edition), https://plato.stanford.edu/archives/win2020/entries/goedel-incompleteness/

1. Gödel said, in a 1939 lecture, that results like the undecidability of first-order logic (see slide 14) show that the human mind could never be replaced by a machine.*
2. Gödel said in a 1951 lecture, that "either … the human mind (even within the realm of pure mathematics) infinitely surpasses the power of any finite machine, or else there exist absolutely unsolvable diophantine problems". Gödel wanted to reject the latter disjunct (Raatikainen 2020).

*I learned this point from Wilfried Sieg, in lectures I attended during the summer of 2010 at CMU.



By unknown - http://www.arithmeum.uni-bonn.de/en/events/285, https://en.wikipedia.org/w/index.php?curid=43426215

Image © National Portrait Gallery. Reproduced here under fair-dealings/the lesson exemption.

Discussion question: The Gödel sentence G says something like "This sentence cannot be proved in the formal system expressing it". By proving Gödel's incompleteness theorem, do we really prove this sentence is true?

3. Two replies to the Gödelian argument

Does Gödel's incompleteness theorem really show that our minds aren't formal machines, akin to a Turing machine? It's helpful to lay out just what the argument is supposed to be …

We're not going to look at Gödel's argument. Recall that while Gödel was sympathetic to the idea, he himself was more cautious. He didn't think the incompleteness theorem alone was enough to show minds aren't machines. We also don't have a written record of him making such a case, so we can't reconstruct it. Instead, the following argument is my attempt to reconstruct what a Gödelian argument against "mechanism" would be. It's based in part on Lucas' influential 1961 paper, a paper that argues incompleteness shows the mind isn't a machine.

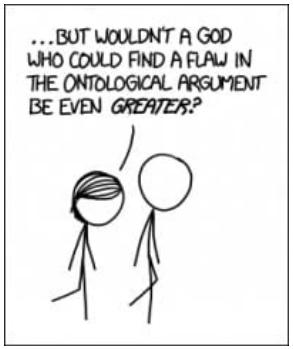
Lucas (1961) "Minds, machines, and Gödel", *Philosophy* 36: 112—27.

Premise 1: If our minds are consistent, self-referentially representable formal machines, then they can't prove their own Gödel sentence G.
Premise 2: Anyone who follows the steps of Gödel's proof can prove any Gödel sentence G, including their own.
Conclusion 1: If our minds are formal machines, then they either aren't consistent or aren't self-referentially representable.
Premise 3: Our minds are consistent and self-referentially representable.
Conclusion 2: Our minds aren't formal machines.

(My reconstruction)

This argument is a lot like Anselm's ontological argument. It's:

- deceptively simple,
- accepted by some people to obviously establish its conclusion, but
- rejected by most people, who all agrees it's obviously flawed, but can't agree on just how to pin down the problem.



From xkcd.com

Note that the argument essentially turns on a simple act of (minor) creativity:

- As premise 2 says, we (human minds) are able to prove a certain theorem. That is, we can *find a proof*, or *discover* something.
- At the same time, as premise 1 says, this would be impossible for a certain sort of machine.

What's super cool about this approach is that (if it works) it will have taken something nebulous and vague — "creativity" — and wrung a mathematically rigorous argument against GOFAI/strong AI out of it!

This argument tries to rigorously prove that we can do something creative that a machine can't!

Does the argument succeed? Lucas and many others thought (think!) so. Turing, who called this argument the "mathematical objection", did not.

(Turing mentioned the issue briefly in his 1950 paper, p. 444, which we covered in week 2.)

Modern critiques of this argument are usually variations on Turing's original critiques, found both in his famous 1950 paper and in other material or wrote or delivered.

For a careful historical discussion, see Piccinini (2003) "Alan Turing and the mathematical objection", *Minds and Machines* 13: 23—48.

As I reconstructed the Gödelian argument above, the argument is valid. (That is, if the premises are true, the conclusions *must* be true.) Hence, the only way to attack the argument is to deny one or more of the premises. Turing focuses on what I've written as premises 1 and 2.

Recall that premise 3 said that our minds are consistent and self-referentially representable. You could try to reject this premise, especially because you might doubt that we're consistent, but I'll set this issue aside. (It seems pretty clear that our minds are capable of self-reference.)

Lucas, in his 1961 paper "Minds, machines, and Gödel" (*Philosophy* 36: 112—27), provides what I take to be a solid argument that while human minds aren't consistent, we're consistent in the way needed for Gödel's argument.

So, let's focus (like Turing) on premises 1 and 2. Recall:

Premise 1: If our minds are consistent, self-referentially representable formal machines, then they can't prove their own Gödel sentence G.
Premise 2: Anyone who follows the steps of Gödel's proof can prove any Gödel sentence G, including their own.

The first thing to notice is that premise 1 isn't *merely* Gödel's incompleteness theorem. Premise 1 actually says a lot more than the incompleteness theorem.

The issue is that the incompleteness theorem is a result about *formal systems*, not *machines*. Granted, a formal system is a set of operations which can be implemented on a machine, but:

1. Machines can implement operations that aren't formal (e.g., a coffee grinder, a Watt Governor, or a connectionist network).
2. Machines are hardware that can potentially run an open-ended range of formal systems ("software").

Photo by Ashkan Forouzani on Unsplash

The phrasing of premise 1 rules out the first worry ("If our minds are … *formal* machines …"), so consider just the second issue (which is what Turing focuses on, anyway).


Photo by Christian Wiediger on Unsplash

As Piccinini (2003, p. 35) explains, Turing asks us to consider a machine that doesn't merely run one program (one formal system), but instead is programmed to stop and try out new rules and axioms (new formal systems).

There's no in-principle reason that this machine, like us, can't expand and find the right new rules and axioms needed to prove Gödel's theorem.

Lucas (1961, pp. 124—26) objects to this reply by saying that while a machine *could* work that way, that's not how conscious human beings do it. He says that humans don't prove Gödel's incompleteness theorem by adding an "extra part", in the way Turing's machine would add extra programing.

But two points:

1. This response seems to beg the question: What reason is there to think we *don't* prove creative new theorems like Gödel's by adding new "parts" (software) to ourselves?
2. Even if we grant Lucas his point, it doesn't seem to help. He's still conceding that a machine can prove its own Gödel sentence is true, just complaining that it doesn't do it "like us".

This brings us to premise 2, which said that anyone who follows the steps of Gödel's proof can prove any Gödel sentence G, including their own.

As we discussed at the end of the last lecture, this premise just seems to be false. What we prove, by proving Gödel's incompleteness theorem, isn't that a formal system's Gödel sentence is true. Instead, what prove is a conditional claim: *If* the system is consistent, *then* its Gödel sentence is true.

The key here is that we only know that a system's Gödel sentence is true if we know that that system is consistent.

The problem is that proving the consistency of any formal system is tricky at best, if not outright impossible. Gödel's own results show that no consistent (self-referentially representable) formal system can be used to prove its own consistency (that's Gödel's *second* incompleteness theorem). So, to prove a system is consistent, you have to use some other system. Well, your proof will then only be good if you know that second system is consistent. Obviously, we've now launched a vicious regress …

Recall the Gödelian argument (as I've reconstructed it):

Premise 1: If our minds are consistent, self-referentially representable formal machines, then they can't prove their own Gödel sentence G.
Premise 2: Anyone who follows the steps of Gödel's proof can prove any Gödel sentence G, including their own.
Conclusion 1: If our minds are formal machines, then they either aren't consistent or aren't self-referentially representable.
Premise 3: Our minds are consistent and self-referentially representable.
Conclusion 2: Our minds aren't formal machines.

The upshot of our discussion is that neither premise 1, nor premise 2, look promising. Both seem to involve confusions, and are probably false.

Discussion question: Let's go back to this idea that not all machines are *formal* machines. Does Gödel's theorem apply to a connectionist (neural) network? Why or why not?