Lecture notes on Gödel's Theorem and Turing's "Mathematical objection".

A) The fundamental idea behind Gödel's theorem

Consider the sentence

> You can't prove this sentence is true

call this sentence G (for Gödel)

- Is G true?

Yes – because if it were false, its opposite would be true. The opposite of G is "You can prove G is true." So if G is false, it's true – this impossible contradiction shows that G cannot be false. So it <u>must</u> be true.

- Can you prove that G is true?

  2 answers:

  Yes – we just did! See the previous bullet point.

  No – G is known to be true, and G states that you can't prove G true. So you can't prove it.

Which answer is right? It depends what you mean by "prove". There are two different types of proof: proving something <u>inside</u> a system using its own rules, or proving something <u>outside</u> a system using external reasoning.

Here, G can be proved true from <u>outside</u> a system, but not from <u>inside</u> a system.

B) Applying the fundamental idea to a Turing test

1. A <u>formal system</u> is a set of rules for transforming statements using a fixed set of characters.

   example: the MIU system in Chapter 1 of GEB.

2. Arithmetic is a formal system.

   e.g. true statements: $6 \times (5+2) = 74$

   $\exists$ means "there exits" $\longrightarrow$ $\exists y \mid 2 \times y = 10$

   $\forall$ means "for all" $\longrightarrow$ $\forall z : 2 \times z = z + z$

   false statements: $3 - 5 = 9$

   $\forall y : y - 1 > 3$

3. (i) Gödel's Theorem (simple version) says:

> G can be translated into arithmetic.
> Therefore, there are statements about numbers which humans know are true, but can't be proved true by the rules of arithmetic
> (Can prove <u>outside</u> the system, not <u>inside</u> it.)

(ii) Gödel's Theorem (more useful version) says:

$$\boxed{\text{Any formal system that includes arithmetic contains statements like G — true statements that can't be proved from within the system.}}$$

note: To find out how to come up with a G-like statement for a given system, you need to read the first half of GEB.
It's fun, but definitely not required for this class.

4. Computers are formal systems.
   - They transform patterns of 0s and 1s according to fixed rules.

5. Computers can do arithmetic
   - Therefore, a computer is a formal system that includes arithmetic
   - Therefore, the 2nd version of Gödel's Theorem, given above, applies to any computer.

6. Therefore, there exists a statement like G for any computer — a statement that we <u>know</u> is true, but which the computer cannot <u>prove</u> using its own rules.

7. Therefore, we can cause a computer to fail a Turing Test as follows. First, the interrogator works out a suitable statement G for the computer opponent. Then the interrogator asks "Is G true?"

A human can work <u>outside</u> the computer's formal system to prove that G is true and thus answer "yes". The computer cannot come up with an answer and is stumped.