6.080 / 6.089 Great Ideas in Theoretical Computer Science
Spring 2008

**6.080/6.089 GITCS** Feb 26, 2008

# Lecture 6

*Lecturer: Scott Aaronson* *Scribe: Tiffany Wang*

## 1 Administrivia

### 1.1 Scribe Notes

If you are doing the scribe notes for a lecture, remind Professor Aaronson to send you his own lecture notes. Lecture 3 has been posted, and Lecture 4 should follow shortly.

### 1.2 Problem Sets/Exams

Pset1 is due this coming Thursday. Submit assignments on class Stellar site or send email to the TA. Typed submissions are preferable. Pset2 will be handed out on Thursday.

Midterm exam will an in-class exam on Thursday, April 3.

## 2 Agenda

Today will be fun! Different structure than previous classes: an open philosophical discussion that will be a good lead into complexity theory. The hope is to motivate more students to get involved in class discussions and to introduce some interesting topics that you should be exposed to at least once in your life.

## 3 Recap

### 3.1 Oracles and Reducibility

Oracles are hypothetical devices that solve a given problem without any computational cost. Assuming the existence of such oracles, we establish a hierarchy of insolvability in which problems may be reducible to one another.

So given two problems A and B, A is reducible to B if there exists a Turing machine M such that $M^B$ solves A, or A $\leq_T$ B.

### 3.2 Turing Degrees

Turing degrees are used to classify all possible problems into groups that are computably equivalent. If given an oracle for one problem in a group, you would be able to solve all other problems of the same Turing degree.

Some examples include the set of all computable problems or the set of problems equivalent to the halting problem, which is incomputable. We also identified that there are problems which are harder than the halting problem: problems that would still be unsolvable even if given an oracle for the halting problem. There also exist problems of intermediate degrees, which reside between the degrees of computable and the halting problem.

### 3.3 Gödel's Incompleteness Theorems

Gödel's theorems are among the top intellectual achievements of last century.

#### 3.3.1 First Incompleteness Theorem

Gödel's First Incompleteness Theorem states: For any fixed formal system of logic F, if the system is sound and computable, then there exist true statements about the integers that are not provable within the system, F. In order to prove those statements you would need a more powerful system, which in turn would also have statements that are not provable and require an even more powerful system, and so on.

Gödel's proof involved a mathematical encoding of the sentence:

$G(F) =$ "This sentence is not provable in $F$."

If $G(F)$ is false, then it is provable, which means $F$ is inconsistent. If $G(F)$ is true, then it is not provable, which means $F$ is incomplete.

#### 3.3.2 Second Incompleteness Theorem

Gödel's Second Incompleteness Theorem states: among the true statements that are not provable within a consistent and computable system F, is the statement of F's own consistency. F can only prove its own consistency if it is inconsistent.

One possible workaround would be to add an axiom to the system which states that F is consistent. However, you would have a new system, F+Con(F), that would not be able to prove Con(F+Con(F)), and so on. You would then establish a hierarchy of more and more powerful theories, each one able to prove consistency of weaker theories but not of itself.

Another proof for Gödel's Incompleteness Theorem is based on the unsolvability of the halting problem. We already established that no Turing Machine exists can solve the halting problem. If we had a proof system that could analyze any Turing machine and prove if it halted or did not halt, we could use this system to solve the halting problem by brute force (also known as the "British Museum algorithm") by trying every possible string that might be a proof. You would either terminate and find a proof that it halts or terminate and find a proof that it doesn't halt. If this proof system was sound and complete, it would violate the unsolvability of the halting problem. Therefore, there is no such sound and complete proof system.

## 4 Completeness vs. Incompleteness

How can we reconcile Gödel's Incompleteness Theorem with his earlier *Completeness* Theorem? Recall that the completeness theorem states: starting from a set of axioms, you can prove anything logically entailed by those axioms by applying the inference rules of first-order logic.

But doesn't this contradict the Incompleteness Theorems? Look, the same guy proved both of them, so there *must* be a resolution! As it turns out, the two theorems are talking about very subtly different things.

The key is to distinguish three different concepts:

1. *True* (assuming the universe we are talking about is the integers)

   The statement is true in the realm of positive integers.

2. *Entailed by the axioms* (true in any universe where the axioms are true)

   This is a *semantic* notion, or based on the meaning of the statements in question. Simply, the statement is true in any situation where the axioms are true.

3. *Provable from the axioms* (provably by applying rules of inference)

   This is a completely mechanical (or *syntactic*) notion, which just means that the statement is derivable by starting from the axioms and then turning a crank to derive consequences of them.

The Completeness Theorem equates provability with entailment. The theorem says that if something is logically entailed by the set of axioms, then it is also provable from the axioms.

The Incompleteness Theorem differentiates entailment from truth over the positive integers. The theorem implies that there is no set of axioms that captures all and only the true statements about the integers. Any set of axioms that *tries* to do so will also describe other universes, and if a statement is true for the integers but not for the other universes then it won't be provable.

### 4.1 Implications

The Incompleteness Theorem was a blow to Hilbert and other mathematicians who dreamed of formalizing all of mathematics. It refuted the belief that every well-posed mathematical question necessarily has a mathematical answer.

However, the question arises of whether incompleteness ever rears its head for any "real" problem. In order to prove his theorems, Gödel had to practically invent the modern notion of a computer, but from a purely mathematical standpoint, his sentences are extremely contrived. So you might wonder what the big deal is! Furthermore, we (standing outside the system) "know" that the Gödel sentence $G(F)$ is true, so who cares if it can't be proved within $F$ itself? (This is a point we'll come back to later in this lecture.)

It took until the 1960's for people to prove that there actually exist mathematical questions that mathematicians wanted answers to, but that can't be answered within the current framework of mathematics.

## 5 Continuum Hypothesis

Recall that Georg Cantor showed there are different kinds of infinity, and specifically that the infinity of real numbers is larger than the infinity of integers.

A related question that Cantor obsessed over (to the point of insanity) until the end of his life was, "Is there any infinity that is intermediate in size between the infinity of real numbers and the infinity of integers?" Cantor formulated the *Continuum Hypothesis* (CH) stating: There is no set whose size is strictly between that of the integers and that of the real numbers.

### 5.1 Gödel and Cohen's results

In 1939, Gödel showed that the Continuum Hypothesis can be assumed consistently. In other words, the Continuum Hypothesis can be assumed to be true without introducing any inconsistency into set theory.

What is set theory? There are different ways of formalizing a set theory, or choosing the right set of axioms to describe a set. The standard form of axiomatic set theory, and the most common foundation of mathematics, is Zermelo-Fraenkel (ZF) set theory.

By Gödel's incompleteness theorems, ZF cannot prove its own consistency, so how could it possibly prove the consistency of itself *plus* the Continuum Hypothesis? Well, it can't! What Gödel proved was a *relative* consistency statement. Namely, if we assume ZF to be consistent, then adding the Contiuum Hypothesis won't make it inconsistent. Conversely, if the Continuum Hypothesis leads to an inconsistency, this can be converted to an inconsistency in ZF itself. Or in logic notation:

$\mathrm{Con(ZF)} \Rightarrow \mathrm{Con(ZF+CH)}$

Then, in 1963, Paul Cohen showed that you can also assume the Continuum Hypothesis is *false* without introducing an inconsistency. In fact, you can insert as many intermediate infinities as you want.

$\mathrm{Con(ZF)} \Rightarrow \mathrm{Con(ZF+\neg(CH))}$

### 5.2 Implications

In George Orwell's novel 1984, the protagonist, Winston Smith, is tortured until he has no will to live. The breaking point is when Smith's assailant is able to get him to admit that 2 + 2 = 5. Orwell is in a way asserting that the certainty of math is a foundation for our beliefs about everything else. So, should we be concerned about the independence of the Continuum Hypothesis? Does it imply that the answers to seemingly reasonable math problems can depend on how we feel about them?

One response is that we ought to step back and ask, when talking about arbitrary subsets of real numbers, whether we really understand what we mean. In some people's view (and Prof. Aaronson would count himself among them), the one aspect of math that we *really* have a direct intuition about is *computation*. So perhaps the only mathematical questions for which we should definite answers are the ones that we can ultimately phrase in terms of Turing machines and whether they halt. There may be other more abstract problems that do have answers (such as the existence of different levels of infinity), but perhaps we should just consider these as added bonuses.

## 6 Thinking Machines

The dream of building a "thinking machine" motivated the creation of formal logic and computer science. On the other hand, to this day there's a huge philosophical debate surrounding what a thinking machine would be and how it would be recognized. A surprisingly large portion of this debate was summarized and even anticipated in a *single paper* written by Alan Turing in 1950, "Computing Machinery and Intelligence".

### 6.1 Turing Test

Turing proposed a criterion called the *Turing Test* to distinguish between humans and machines. If a human interacting with a machine cannot reliably distinguish it from a human, then the machine

ought to be regarded as intelligent, just as the human would be.
*Response:* But it's just mechanical contrivance! *Clearly* it's not really conscious like I am; it doesn't really have feelings.
*Response to the response:* Set aside yourself and think about other people. How can you be certain that *other people* are conscious and have feelings?

You infer the consciousness of a person based on interactions. Likewise, if a computer program interacted with you in a way that was indistinguishable from how a person would, you should be willing to make the same inference.

Perhaps Turing himself said it best:

> [Solipsism] may be the most logical view to hold but it makes communication of ideas difficult. A is liable to believe "A thinks but B does not" whilst B believes "B thinks but A does not." Instead of arguing continually over this point it is usual to have the polite convention that everyone thinks.

*Question from the floor:* Can humans fail the Turing Test?

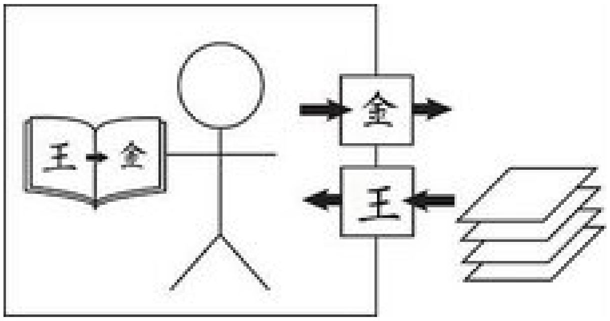
Great question! The Loebner Prize is an annual competition that awards prizes to the most humanlike Chatterbox. On the subject of Shakespeare, a librarian was repeatedly judged to be a machine because people did not believe a human could possibly know so much about Shakespeare.

Many people have argued that passing the Turing Test is a *sufficient but not necessary* condition for intelligence.

## 6.2 Searle's Chinese Room

Searle's Chinese Room is a thought experiment designed by John Searle (1980) in response to the Turing Test. Searle wanted to dramatize the point that carrying out computations and manipulating symbols does not constitute real awareness or intelligence.

*Searle:* Suppose you sealed me in a room and fed me slips of paper with Chinese characters written on them, and suppose I had a giant rulebook for producing other slips of Chinese characters, constituting a fluent response to you. By exchanging these slips of paper, I could simulate a Chinese conversation without actually knowing Chinese. Therefore simple symbol manipulation does not constitute understanding.[1]



What's a possible response to this argument?
*System Response:* The problem with Searle's argument is the need to distinguish between Searle and the system consisting of him and the rulebook. Searle may not understand Chinese, but the system as a whole does understand.

[1] It's a strange experience to explain Searle's thought experiment to students many of whom *would* understand what was on the slips! –SA

*Searle:* That's ridiculous! (In his writings, Searle constantly appeals to what he regards as common sense.) Just memorize the rulebook, thereby removing the system.
*Response:* You would then have to distinguish between Searle and the person being simulated by his memory.

Another response is Searle gets a lot of the mileage in his thought experiment from careful choice of imagery: "mere slips of paper"! However, the human brain has immense computational capacity (roughly $10^{11}$ neurons and $10^{14}$ synapses, with each neuron itself much more complex than a logic gate), and performs its computations massively in parallel. Simulating the computational power of the brain could easily require enough slips of paper to fill the solar system. But in that case Searle's scenario seems to lose its intuitive force.

## 6.3 Competing Analogies

The debate surrounding the feasibility of truly thinking machines comes down to competing analogies.

The central argument against the possibility of intelligent machines has always been that "A computer simulation of a hurricane doesn't make anyone wet." But on the other hand, a computer simulation of multiplication clearly *is* multiplication.

So the question boils down to: is intelligence more like a hurricane or like multiplication?

## 6.4 The "Practical" Question

Setting aside whether or not a machine that passes the Turing Test should be considered conscious, there's also the more "practical" question: can there ever *be* a machine that passes the Turing Test? Or is there some fundamental technological limitation?

In 1950, Turing predicted that by the year 2000, a machine would be able to fool the average person 70% of the time into thinking it was human after a 5-minute conversation. The accuracy of his prediction depends on the sophistication of the "average person" judging the machine.

Already in the 1960s, computer chat programs were easily able to fool unsophisticated people. Chat programs like ELIZA (or more recently AOLiza) are based on parroting the user like a "psychotherapist": "Tell me more about your father." "I would like to go back to the subject of your father." People would pour their hearts out to these programs and refuse to believe they were talking to a machine.

Sophisticated people who know the right thing to look for would easily uncover the identity of the program by asking any commonsense question: "Is Mount Everest bigger than a shoebox?" The program would answer with something like "Tell me more about Mount Everest," and continue parroting the person instead of giving a straight answer.

A current practical issue involves CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart). A CAPTCHA is a test that current computers can generate and grade but cannot pass. These tests were invented to block spambots but actually raise a profound philosophical issue: that of distinguishing between humans from machines.

There is an arms race between the spammers and the CAPTCHA-makers. Some of the commonly used CAPTCHA's have been broken, but in general the CAPTCHA-makers are still winning.

# 7 Gödel and Thinking Machines

The idea that Gödel's Incompleteness Theorem somehow proves the impossibility of thinking machines is an old one (indeed, Gödel himself might have believed something close to this). Today, though, the idea is most closely associated with Roger Penrose, the famous mathematical physicist who, among other achievements, invented Penrose tiles and (along with Stephen Hawking) showed that General Relativity generically predicts black holes.

## 7.1 The Emperor's New Mind

In 1989, Penrose wrote a book, *The Emperor's New Mind*, in which he tried to use Gödel's Incompleteness Theorem to argue that computers would never be able to simulate human beings. Consider again the Gödel sentence

G(F) = "This sentence is not provable in F."

Penrose argues that any computer working within the logical system F cannot prove G(F), but we as humans can just "see" that it is true by engaging in meta-reasoning. Therefore humans can do something that computers can't.

*Question from the floor:* Can you change the statement to "This sentence cannot be proved by Roger Penrose?"?

Great question! One possible response is that the modified sentence can't be compiled into a purely logical form, since we don't yet completely understand how the human brain works. This is basically an argument from ignorance.

*Another response:* Why does the computer have to work within a fixed formal system F?

*Penrose:* Because otherwise, the computer would not necessarily be sound and could make mistakes.

*Response:* But humans make mistakes too!

*Penrose:* When I perceive G(F) is true, I'm absolutely certain of it.

But how certain are we that G(F) is true? Recall that Con(F) (the consistency of F) implies G(F).

Claim: The inverse is true as well. That is, G(F) implies Con(F).

Proof: If F is inconsistent, then it can prove anything, including G(F). Hence $\neg$ Con(F) implies $\neg$ G(F) (i.e., that G(F) is provable), which is the contrapositive of what we wanted to show.

So the bottom line is that G(F) is simply equivalent to the consistency of F. The question, then, is whether or not human beings can step back and "directly perceive" the consistency of a system of logic, which seems like more of a religious question than a scientific one. In other words, one person might be absolutely certain of a system's consistency, but how could that person ever convince someone else by verbal arguments?

## 7.2 Views of Consciousness

Penrose devised a classification of views about consciousness:

1. Simulation produces consciousness. (Turing)

2. Consciousness can be simulated, but mere simulation does not produce consciousness. (Searle)

3. Consciousness cannot even be simulated by computer, but has a scientific explanation. (Penrose)

4. No scientific explanation. (99% of people)