The Fundamental Theorem

The reason Turing Machines are so incredibly useful is that it is possible to prove the following theorem:

> **Fundamental Theorem of Turing Machines**
> A function from natural numbers to natural numbers is Turing-computable if and only if it can be computed by an ordinary computer, given unlimited memory and running time.

(What I mean by "ordinary computer" here is what sometimes gets referred to as a "register machine", but the details won't matter for present purposes.)

The proof of the Fundamental Theorem is long and clunky, but the basic idea is straightforward:

- One shows that every Turing-computable function is computable by an ordinary computer (given unlimited memory and running time) by showing that one can program an ordinary computer to simulate any given Turing Machine.
- One shows that every function computable by an ordinary computer (given unlimited memory and running time) is Turing-computable by showing that one can find a Turing Machine that simulates any given ordinary computer.

In fact, computer scientists tend to think that something stronger than the Fundamental Theorem is true:

> **Church-Turing Thesis**
> A function is Turing-computable if and only if it can be computed algorithmically.

For a problem to be solvable **algorithmically** is for it to be possible to specify a finite list of instructions for solving the problem such that:

1. following the instructions is guaranteed to yield a solution to the problem, in a finite amount of time.

2. the instructions are specific enough that they could be carried out by a fully autonomous mechanism whose workings we fully understand.

Intuitively speaking, you can think of the Church-Turing Thesis as stating that a function is Turing-computable if and only if there is a *finite* way of specifying what its values are. So, in particular, a function that fails to be Turing-computable is a function that is so complex that its values cannot be finitely specified.

I mentioned above that computer scientists tend to think that the Church-Turing Thesis is true. But this is not because they are able to prove it. It's actually not clear what a formal proof of the Church-Turing Thesis would look like. The problem is that such a proof would require a formal characterization of the notion of an algorithmic computability, and it is not clear that one could formalize the notion of algorithmic computability without thereby restricting its scope. (Notice, for example, that the notion of Turing-computability is itself one natural way of formalizing the notion of algorithmic computability. On such a formalization, the Church-Turing Thesis is certainly true. But it is also trivial.)

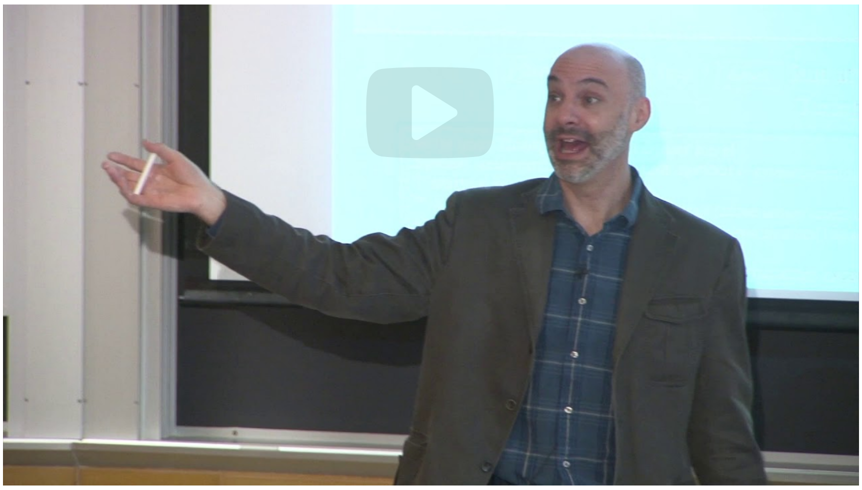
The reason the Church-Turing Thesis is widely regarded as true is that any sensible formalization of the notion of algorithmic computability that anyone has ever been able to come up with is provably equivalent to the notion of Turing-computability.

Programming with Turing Machines can be somewhat cumbersome. But Turing Machines are so incredibly simple that theorizing about Turing Machines tends to be pretty straightforward. In light of the Fundamental Theorem, this means that theorizing *about* Turing Machines is an extremely useful way of gaining insight about computers more generally.

Assuming the Church-Turing Thesis is true, it also means that theorizing about Turing Machines is an extremely useful way of gaining insight about algorithmic methods more generally.

Video Review: Church-Turing Thesis

Start of transcript. Skip to the end.



0:00 / 0:00 1.50x

The fact that they are so simple means two things.

First, it means that programming anything

interesting on a Turing Machine is an extraordinary pain.

So if you actually want to program things,

you should not use a Turing machine.

You should use C++ or

Video

Download video file

Transcripts

Download SubRip (.srt) file

Download Text (.txt) file

Discussion

Hide Discussion

Topic: Week 9 / The Fundamental Theorem

Add a Post

Show all posts by recent activity

There are no posts in this topic yet.