An interpreter (or compiler) is a program that operates on programs. In fact, there are numerous other ways to operate on programs. For example, given a one-parameter function in some language, produce the function that computes its derivative.

The development of program-analysis tools of this sort is an active research area.

### The Halting Problem

For example, would be very useful to know "Is there some input to Scheme function \(P\) that will cause it to go into an infinite loop?" Is there a program that operates on programs that will answer this question correctly in finite time?

This question was answered negatively in the 1930s by Alan Turing. In fact, there isn't even a program that fully meets the following specification:

```scheme
(define (halts? defn x) ...)
```

### Biting Your Tail: Proof of Impossibility

```scheme
(define halts?-bogus-program
  (quote
    (define (halts?-bogus x)
      (define (halts? defn x)
        (alleged definition of halts?))
      (define (loop) (loop))
      (if (halts? x x) (loop) #t)))

(halts? halts?-bogus-program halts?-bogus-program) ; (*)
```

Assume that `halts?` works as specified: `(halts? defn y)` returns true if `defn` is a Scheme definition of some one-argument function that halts (does not loop) when given input `y`. Then if the line marked (*) returns true, it is supposed to mean that `halts?-bogus` halts.

But `halts?-bogus` computes `(halts? x x)` during its execution, with the value of `x` being `halts?-bogus-program`. That would presumably return true, which would make `halts?-bogus` loop infinitely.

So clearly, if `halts?` works, line (*) cannot return true after all; it must return false.

### Biting Your Tail (II)

```scheme
(define halts?-bogus-program
  (quote
    (define (halts?-bogus x)
      (define (halts? defn x)
        (alleged definition of halts?))
      (define (loop) (loop))
      (if (halts? x x) (loop) #t)))

(halts? halts?-bogus-program halts?-bogus-program) ; (*)
```

But if the line marked (*) returns false, then the execution of `halts?-bogus` would terminate, which would mean that `halts?` had gotten the wrong answer.

The only way out is to conclude that `halts?` never returns in this case—it does not answer the question for all possible inputs.

Putting it all together, we must conclude that

* No possible definition of `halts?` works all the time.

Therefore, the impossibility of the halting problem is fundamental: the `halts?` function is uncomputable.

If `halts?` always returns a correct result (when it returns), then there must be an infinite number of inputs for which it fails to give any answer at all (i.e., loops infinitely). Why infinite?

### Consequences

There's a lot of fallout from the impossibility of writing `halts?`.

For example, I cannot tell in general whether two programs compute the same thing. Therefore, Perfect anti-virus software is theoretically impossible.

Anti-virus software must either miss some viruses, or prevent some innocent programs from running (or freeze your computer.)

Many analyses that might be useful cannot be done in general. For example, even if I know that a given program will terminate, I cannot necessarily predict in general how long it will take to do so.
Axioms:

Big Idea:

For our "theory of..." we can add enough constraints to get the properties

Meaning from Assertions

From Syntax to Semantics

The Mathematics of Mathematics

Formal Systems
Can all the true things be proven? That we can't know whether some statements whether we want them to. At which point, all statements are provable, which is impossible. Inconsistent (with respect to the axioms), which is inconsistent. (With respect to other axioms, which is consistent.)

The Incompleteness Theorem might seem to say that the latter is impossible. Therefore: Any consistent mathematical system

There is but one way out: "valid" doesn't mean what we think. Instead, it means that a formula can be proven. The Incompleteness Theorem might seem to say that the latter is impossible. Therefore: Any consistent mathematical system

The completeness theorem says that any consistent mathematical system is complete:

The completeness theorem says that any consistent mathematical system is complete:

Nonstandard Models

Nonstandard Models

The incompleteness theorem might seem to say that the latter is impossible. Therefore: Any consistent mathematical system

The completeness theorem says that any consistent mathematical system is complete:

The completeness theorem says that any consistent mathematical system is complete:

The incompleteness theorem might seem to say that the latter is impossible. Therefore: Any consistent mathematical system

The completeness theorem says that any consistent mathematical system is complete:

The completeness theorem says that any consistent mathematical system is complete:

The incompleteness theorem might seem to say that the latter is impossible. Therefore: Any consistent mathematical system

The completeness theorem says that any consistent mathematical system is complete:

The completeness theorem says that any consistent mathematical system is complete:

The incompleteness theorem might seem to say that the latter is impossible. Therefore: Any consistent mathematical system

The completeness theorem says that any consistent mathematical system is complete:

The completeness theorem says that any consistent mathematical system is complete:

The incompleteness theorem might seem to say that the latter is impossible. Therefore: Any consistent mathematical system

The completeness theorem says that any consistent mathematical system is complete:

The completeness theorem says that any consistent mathematical system is complete:

The incompleteness theorem might seem to say that the latter is impossible. Therefore: Any consistent mathematical system

The completeness theorem says that any consistent mathematical system is complete:

The completeness theorem says that any consistent mathematical system is complete:

The incompleteness theorem might seem to say that the latter is impossible. Therefore: Any consistent mathematical system

The completeness theorem says that any consistent mathematical system is complete:

The completeness theorem says that any consistent mathematical system is complete:

The incompleteness theorem might seem to say that the latter is impossible. Therefore: Any consistent mathematical system

The completeness theorem says that any consistent mathematical system is complete:

The completeness theorem says that any consistent mathematical system is complete:

The incompleteness theorem might seem to say that the latter is impossible. Therefore: Any consistent mathematical system

The completeness theorem says that any consistent mathematical system is complete:

The completeness theorem says that any consistent mathematical system is complete:

The incompleteness theorem might seem to say that the latter is impossible. Therefore: Any consistent mathematical system

The completeness theorem says that any consistent mathematical system is complete:

The completeness theorem says that any consistent mathematical system is complete:

The incompleteness theorem might seem to say that the latter is impossible. Therefore: Any consistent mathematical system

The completeness theorem says that any consistent mathematical system is complete:

The completeness theorem says that any consistent mathematical system is complete: