The Definition of Algorithm

The idea behind digital computers may be explained by saying that these machines are intended to carry out any operations which could be done by a human computer.

(Alan Turing)

The Church-Turing Thesis

Informally, an algorithm is a collection of simple instructions for performing some task.

The Church-Turing thesis proposes that the intuitive notion of algorithms is equivalent to Turing machines.

The evidence is that all known algorithmic languages are equivalent to Turing machines.

Implies that Turing-recognizable languages are the languages recognizable by some algorithm, and decidable languages are the languages decidable by some algorithm.

Example: Hilbert’s Tenth Problem

Can an algorithm determine if a polynomial with integer coefficients has integer roots?

Here is a “Turing machine” for this problem:

1. The input is a polynomial \( p \).
2. a. Enumerate assignments to the variables,  
2. b. and evaluate \( p \) for each assignment. 
3. Accept whenever \( p \) evaluates to 0.
Example Continued

$L = \{p \mid p \text{ is a polynomial with integer coefficients } \land p \text{ has an integer root}\}$

$L$ is Turing-recognizable, but not decidable.
Conclude $L$ is not decidable in any programming language.

$L_1 = \{p \mid p \text{ is a polynomial with integer coefficients } \land p \text{ has one variable } \land p \text{ has an integer root}\}$

$L_1$ is Turing-recognizable and decidable.
Conclude $L_1$ is decidable in any general programming language.

Describing Turing Machines

The input to a Turing machine is a string.
Should show how to represent any input as a string.

Because of the Church-Turing thesis, any sequence of clear instructions is acceptable for describing an algorithm.

However, the instructions should apply to how the input is represented as a string.