Session 35: Halting Problem

- Definition of Halting Problem
- A Famous Theorem

Unsolvable Problems

Can every problem by solved by an algorithm?

Answer (Turing): No!

He defined an unsolvable problem, the **halting problem**:
Can we develop a procedure that takes as input a computer program along with its input and determines whether the program will eventually halt with that input?

Halting Problem

Theorem: The halting problem that cannot be solved using any procedure.

The proof requires an accurate description of what is a procedure, the input and output and of how a procedure can be encoded as string (Turing machine).

Proof Sketch

Assume that there is such a procedure and call it *H*(*P*, *I*).

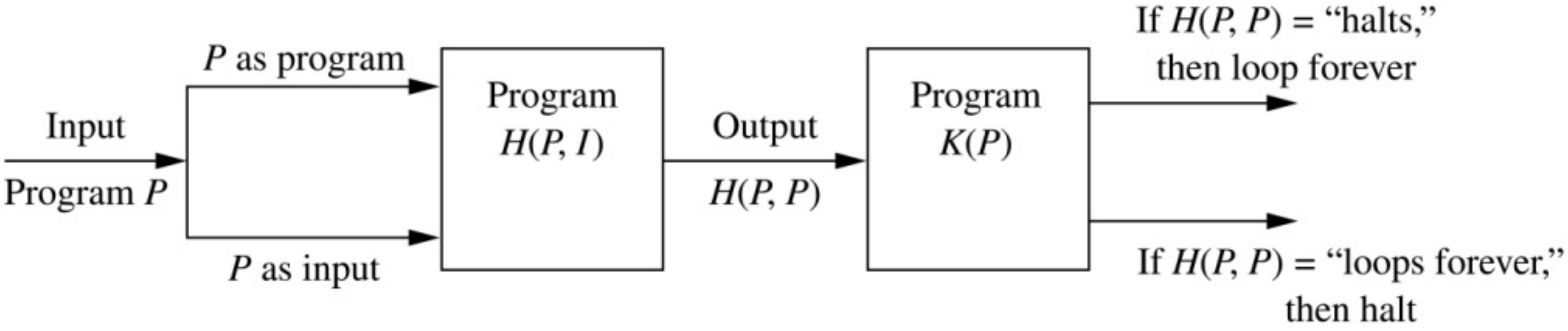
The procedure *H*(*P*, *I*) takes as input a program *P* and the input *I* to *P*.

- H outputs “halt” if it is the case that *P* will stop when run with input *I*.
- Otherwise, *H* outputs “loops forever.”

Construct a procedure *K*(*P*), which works as follows.

- If *H*(*P*, *P*) outputs “loops forever” then *K*(*P*) halts.
- If *H*(*P*, *P*) outputs “halt” then *K*(*P*) goes into an infinite loop

Proof Sketch



Now we call *K* with *K* as input, i.e. *K*(*K*).

- If the output of *H*(*K*, *K*) is "loops forever" then *K*(*K*) halts. A Contradiction.
- If the output of *H*(*K*, *K*) is "halts" then *K*(*K*) loops forever. A Contradiction.

Therefore, there can not be a procedure that can decide whether or not an arbitrary program halts.

Summary

- Concept of Algorithm
- Searching and Sorting Algorithms
- Greedy algorithms take locally the best decisions
- Algorithms can show the existence of a solution to a problem
- Not every problem can be solved by an algorithm