

The Halting Problem

The Halting Problem

Given: source code for a program P and x an input we could give to P
Return: True if P will halt on x , False if it runs forever (e.g. goes in an infinite loop or infinitely recurses)

This would be super useful to solve!

We can't solve it...let's find out why.

A Very Tricky Program.

```
Diagonal.java(String x){
    Run H.exe on input <x, x>
    if(H.exe says "x halts on x")
        while(true){//Go into an infinite loop
            int x=2+2;
        }
    else //H.exe says "x doesn't halt on x"
        return; //halt.
}
```

A Reduction

```
Trick(P, x) {  
  Run P on x, //(but only simulate printing if P prints things)  
  Print "Hello World"  
}
```

This actually prints "hello world" iff P halts on x.

Plug Trick into W and....we solved the Halting Problem!

Reductions in General

The big idea for reductions is "reusing code"

Just like calling a library

But doing it in contrapositive form.

Instead of

"If I have a library, then I can solve a new problem" reductions do the contrapositive:

"If I can solve a problem I know I shouldn't be able to, then that library function can't exist"