Another undecidable problem

1’s problem: Given the code of a program $M$ does $M$ output 1 on input 1? If so, answer 1 else answer 0.

Claim: the 1’s problem is undecidable

Proof: by reduction from the Halting Problem

What we want for the reduction

- Halting problem takes as input a pair $<P,x>$
- 1’s problem takes as input $<M>$
- Given $<P,x>$ can we create an $<M>$ so that $M$ outputs 1 on input 1 exactly when $P$ halts on input $x$?

Yes

- Here is all that we need to do to create $M$
  - modify the code of $P$ so that instead of reading $x$, $x$ is hard-coded as the input to $P$ and get rid of all output statements in $P$
  - add a new statement at the end of $P$ that outputs 1.
- We can write another program $T$ that can do this transformation from $<P,x>$ to $<M>$

How we might do the hard-coding if the code were in C?

- Include an assignment at the start that would place the characters in string $x$ in some array $A$.
- Replace all scanf’s in $P$ with calls to a new function scanfA that simulates scanf but gets its data from array $A$.
- Replace all printf’s in $P$ by printB which doesn’t actually do anything.

Finishing things off

- Therefore we get a reduction
  - Halting Problem $\leq 1$’s problem
- Since there is no program solving the Halting Problem there must be no program solving the 1’s problem.
Why the name reduction?

- Weird: it maps an easier problem into a harder one
- Same sense as saying Maxwell reduced the problem of analyzing electricity & magnetism to solving partial differential equations
- Solving partial differential equations in general is a much harder problem than solving E&M problems

A geek joke

- An engineer
  - is led in a kitchen with an empty kettle on the table and told to boil water; she fills the kettle with water, puts it on the stove, turns on the gas and boils water.
  - she is next confronted with a kettle full of water sitting on the counter and told to boil water; she puts it on the stove, turns on the gas and boils water.
- A mathematician
  - is led in a kitchen with an empty kettle on the table and told to boil water; she fills the kettle with water, puts it on the stove, turns on the gas and boils water.
  - he is next confronted with a kettle full of water sitting on the counter and told to boil water: he empties the kettle in the sink, places the empty kettle on the table and says, "I've reduced this to an already solved problem!"

A general phenomenon: Can’t tell a book by its cover

- Suppose you have a problem $A$ that asks given program code $<P>$, to determine some property of the input-output behavior of $P$, answering 1 if $P$ has the property and 0 if $P$ doesn’t have the property.
- Rice’s Theorem: If $A$’s answer isn’t always the same then there is no program deciding $A$

Even harder problems

- Recall that with the halting problem, we could always get at least one of the two answers correct
  - if it halted we could always answer 1 (and this would cover precisely all 1’s we need to do) but we can’t be sure about answering 0
- There are natural problems where you can’t even do that!
  - e.g. Given the codes of two programs, $P$ and $Q$, answer 1 if they compute the same function and 0 if they compute different functions

Quick lessons

- Don’t rely on the idea of improved compilers and programming languages to eliminate major programming errors
  - truly safe languages can’t possibly do general computation
- Document your code!!!!
  - there is no way you can expect someone else to figure out what your program does with just your code ....since....in general it is provably impossible to do this!