# Last time

- Z3
- Proving stuff about programs!
  - super powerful
  - super cool

# Coming up

- Final projects:
  - final project presentations: Tue Dec 12, in CS 150
  - final submission due: Fri Dec 15, 11:55 PM

# Project Final Presentations

- Next Tuesday (Dec 12) 10AM-11:15AM
- CS 150 (in the CS building)
- Think of this as a science fair.
- Each team will get an easel. Bring a poster or printed slides. And laptop for demo.
- Describe and discuss the solution, and demo the implementation.
- Will see (at least) 2 separate judges.
- Chance to see other projects too!

# Today’s plan

- Evaluations
- Power of software

# Evaluations

- We'll take 15 minutes to do evaluations

- They are anonymous and I don't see them until (long) after the grades are posted
- I actually use them to improve my teaching
- UMass uses them to decide if I am a good teacher

# Evaluations

http://owl.oit.umass.edu/partners/courseEvalSurvey/uma/

- If we get 80% participation by tomorrow:
  - Everyone gets 2 points of extra credit.
  - Everyone gets a chance to submit an optional extra credit assignment.

# Power of Software

Can you write any program I describe to you?

# Can you write:

A program HALTS? whose input is the body of a method, and that outputs false if the method enters an infinite loop, and true if it does not.

# What's HALTS?(method)?

```
method() {
  print "hello world";
}
```

# What's HALTS?(method)?

```
method() {
   for (int x=0; x<5; x++)
      print "hello world";
}
```

# What's HALTS?(method)?

```
method() {
   for (int x=0; x<-1; x++)
      print "hello world";
}
```

# What's HALTS?(method)?

```
method() {
  while (true);
}
```

# What's HALTS?(method)?

```
method() {
   int x = 785th digit of π;
   if (x == 7)
      while(true);
}
```

# What's HALTS?(method)?

```
method() {
   int x = 785th digit of π;
   int y = x^x^x^x^x+1;
   int z = yth digit of π;
   if (z == 0)
      while(true);
}
```

# What's HALTS?(method)?

```
method() {
   int x = 785th digit of π;
   int y = x^x^x^x^x+1;
   int[] z[] = the yth through (x+y)th
                  digits of π;
   if (z ever repeats in π again)
      while(true);
}
```

# How about the general case?

- Let’s count programs.  How many programs are there?

# Specifications

- And how many specification are there?
  - let's limit ourselves to simple specifications:
    - given a set of numbers, e.g., {2, 4, 6}
    - on input i, return 1 if i is in the set, and 0 otherwise

# First 64 programs

- How many of our specifications can I solve with 64 programs?

  (a) 64

  (b) 32

  (c) 8

  (d) 6

  (e) 2

# First 64 programs

- With 64 programs, how large can my specification sets get (if I am being compact)
  (a) 64
  (b) 32
  (c) 8
  (d) 6
  (e) 2
- Example: with 4 programs, I could cover: {}, {1}, {2}, {1,2}

# Scalability Problem

- To cover subsets of a set of n numbers, I need $2^n$ programs.
- But I only have as many programs are there are natural numbers.
- That's exponentially smaller than the number of specifications there are.

Can't do it for all subsets!

# Can HALTS? exist?

- Imagine that you wrote HALTS?
- I will write a new program NALTS?:

```
NALTS?(Method p) {
   if (HALTS?(p)==false) return 1;
   else while (true);
}
```

Key: run the program on itself

What is the value of

NALTS?(NALTS?)

# What is the value of NALTS?(NALTS?)

- Two cases:

1. If NALTS?(NALTS?) goes into an infinite loop, then
   HALTS?(NALTS?)==true, which means that NALTS? terminates.
   So case 1 is impossible.
2. If NALTS?(NALTS?) does not go into an infinite loop, then HALTS?(NALTS?)==false, which means that NALTS? does not terminate.
   So case 2 is impossible.

# Conclusion

- The program HALTS cannot exist!
- Many programs cannot exist!

- Learn more in CS 401 or CS 601

# Zero-Knowledge Proofs

How can I prove to you I know X without telling you anything about X?