What computers just *cannot* do.

COS 116: 3/1/2007

Adam Finkelstein

Administrivia

- Do we want “precepts” at start of lab?
- Take-home midterm in midterms week
  (closed book, 3-hour test). Preferences?
- Couple review sessions before midterm in the evening.
- Handouts today: 2 articles; HW 2

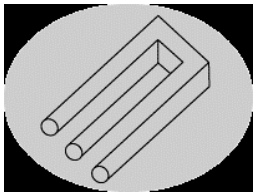
“Prof, what’s with all the negative thinking?!?”

- An obvious motivation:
  Understand the limits of technology

The power of negative thinking….

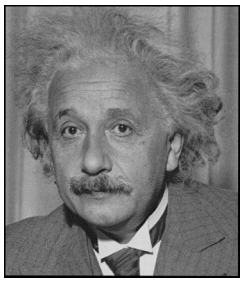
History-of-science perspective

Often, impossibility result → deep insight



Examples

- Impossibility of trisecting angle with ruler and compass (Galois) → Group Theory and much of modern math



- Nothing travels faster than light → Relativity and modern physics

CAPTCHA & ESP

[Luis von Ahn]

Computer generated test that computers cannot pass easily.

Reminiscent of cryptography

Image labeling

Can mathematicians be replaced by machines? [Hilbert, 1900]

Math is axiomatic

Axioms – Set of statements

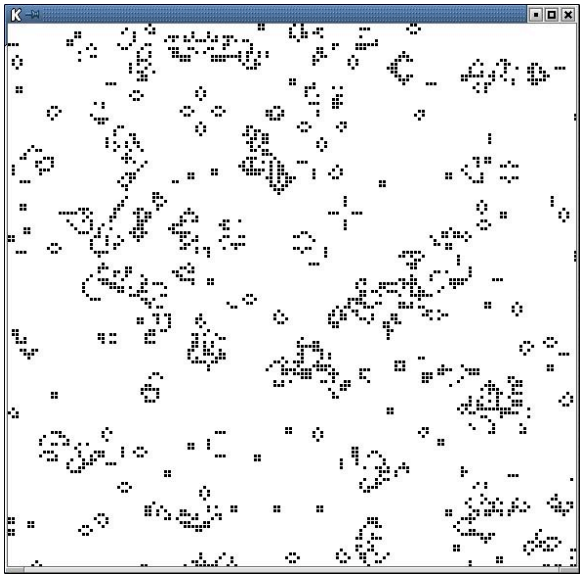
Derivation rules – finite set of rules for deriving new statements from axioms

Theorems – Statements that *can* be derived from axioms in a finite number of steps

Mathematician – Person who tries to determine whether or not a statement is a theorem.

Can a simple set of mathematical equations “solve” problems like:

Given starting configuration for the game of life, determine whether or not cell (100,100) is ever occupied by a critter.



John Conway

Automated Checking of Software?

Windows XP: 40 million line program

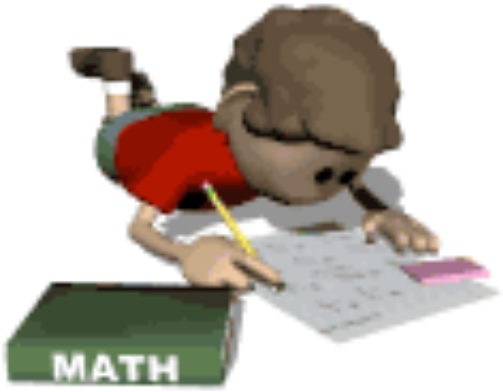
Can computers check whether or not it will ever crash?

What is computation?

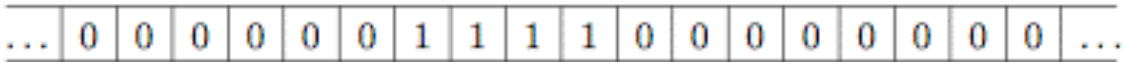
A formalization of an age-old notion

Basic Elements

- **Scratch Pad**
- **Step-by-step description of what to do ("program"); should be finite!**
- **At each step:**
  - **Can only scan a fixed number of symbols**
  - **Can only write a fixed number of symbols**



Turing's model



- 1 dimensional unlimited scratchpad ("infinite")
- Only symbols are 0/1 (tape initially has all 0s.)
- Can only scan/write one symbol per step
- Program looks like →

1. PRINT 0
2. GO LEFT
3. GO TO STEP 1 IF 1 SCANNED
4. PRINT 1
5. GO RIGHT
6. GO TO STEP 5 IF 1 SCANNED
7. PRINT 1
8. GO RIGHT
9. GO TO STEP 1 IF 1 SCANNED
10. STOP

The Doubling Program

What does this program do?

1. PRINT 0
2. GO RIGHT
3. GO TO STEP 1 if 1 SCANNED
4. GO TO STEP 1 if 0 SCANNED

Turing –Church Thesis

This model exactly captures what computation is.

It can simulate every other computational model that can be physically built.

“Code” for a program

= Binary Representation

Many conventions possible (e.g., ASCII)

Davis’s convention:

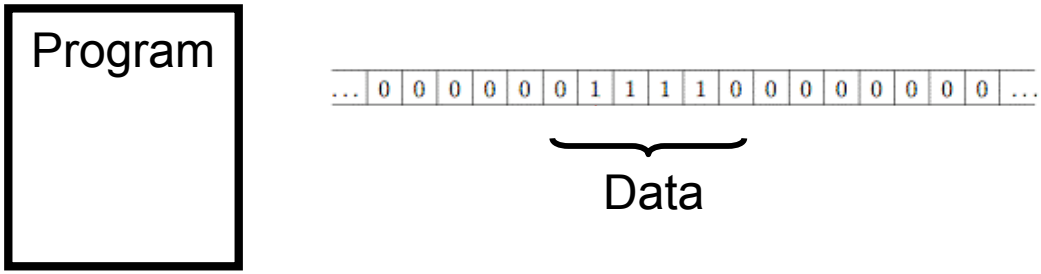
Code	Instruction
000	PRINT 0
001	PRINT 1
010	GO LEFT
011	GO RIGHT
$1010\underbrace{\ldots}_{i}01$	GO TO STEP i IF 0 IS SCANNED
$1101\underbrace{\ldots}_{i}10$	GO TO STEP i IF 1 IS SCANNED
100	STOP

P ⟷ Code (P)

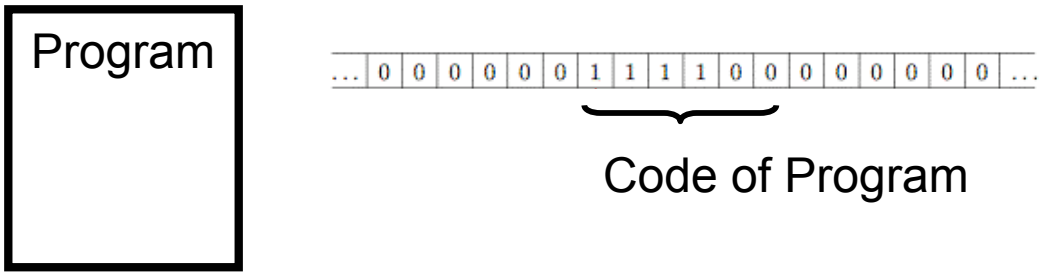
Programs and Data

A False Dichotomy

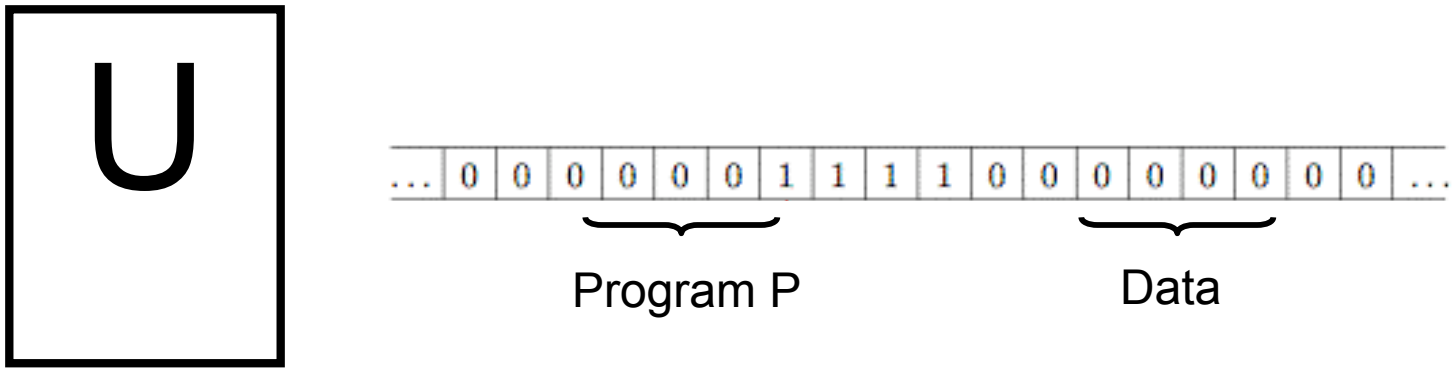
Usual viewpoint -



But can have -



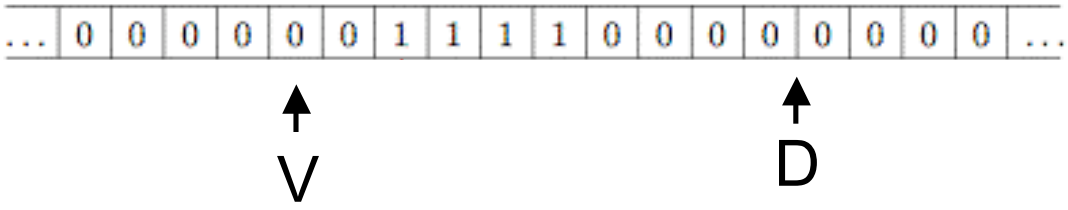
Universal Program U



- U “simulates” what P would do on that data

Automated Bug Checking Revisited

Halting Problem



Let P = program such that code(P) = V.
Does P halt on data D?

Trivial Idea: Simulate P using universal program U.
If P halts, will eventually detect.

Problem: But if P never halts, neither does the simulation.

Next Time: Halting Problem is unsolvable by another program

Also, some class discussion of the readings.

Need to understand notion of Turing-Post program (e.g., doubling program) and what a universal program is.