Topics for Discussion

1. How did the Fowler and Jeon apply the insights from the CLEVER algorithm in studying Supreme Court decisions and what did they learn?

2. What did you learn about data mining from the readings and what reactions does that provoke in you?

Administrivia

- “Precepts” will be Mon 6:15-7pm and Tues 6:15-7pm in lab

- Take-home midterm in midterms week (closed book, 3-hour test). Preferences?

- Couple of review sessions before midterm in the evening.

- Handouts today: 2 articles; Lab 5; HW 2

What can computers not do?

3/2/2006

COS 116

Instructor: Sanjeev Arora

What's with the negative thinking, Prof.?

- An obvious motivation: Understand the limits of technology

The power of negative thinking….

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

Math is axiomatic

Axioms – Set of statements

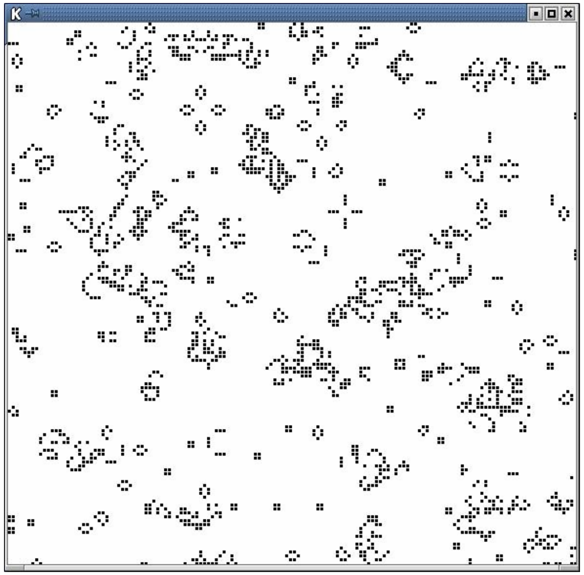
Derivation rules – finite set of rules for deriving new statements from axioms

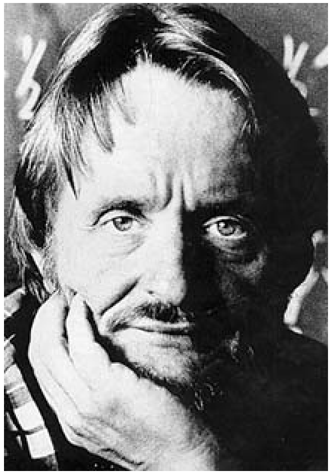
Theorems – Statements that *can* be derived from axioms in a finite number of steps

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 a starting configuration for 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 we use computers to check whether or not it will ever crash?

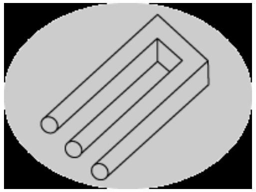
CAPTCHA

Computer generated test that current computers cannot pass easily.

Also reminiscent of cryptography

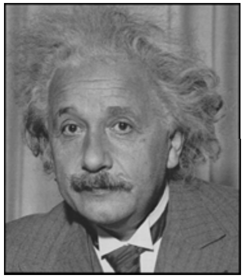
History-of-science perspective

Often, impossibility result ⟶ deep insight



Examples

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



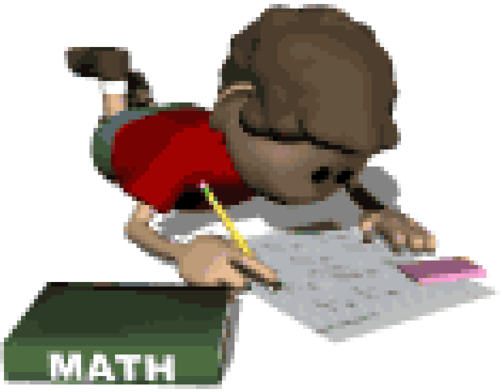
- Discovery that nothing travels faster than light ⟶ Relativity and modern physics

What is computation?

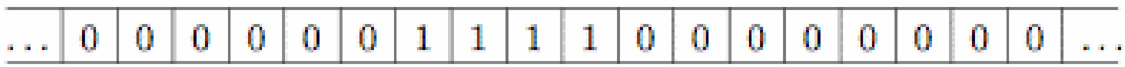
A formalization of an age-old notion

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)

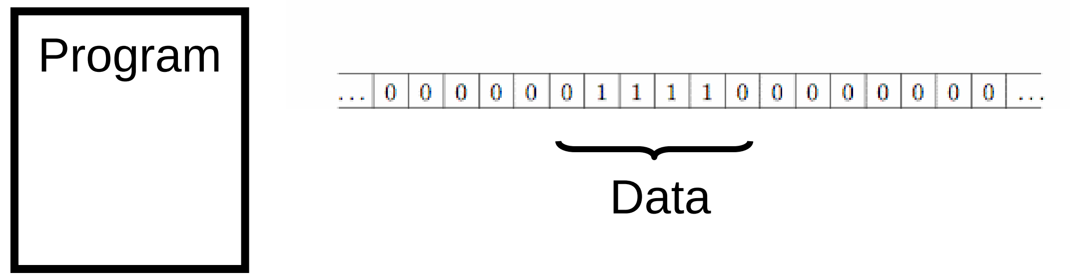
Davis’s convention:

Code	Instruction
000	PRINT 0
001	PRINT 1
010	GO LEFT
011	GO RIGHT
$101\underbrace{0\ldots0}_{i}1$	GO TO STEP i IF 0 IS SCANNED
$110\underbrace{1\ldots1}_{i}0$	GO TO STEP i IF 1 IS SCANNED
100	STOP

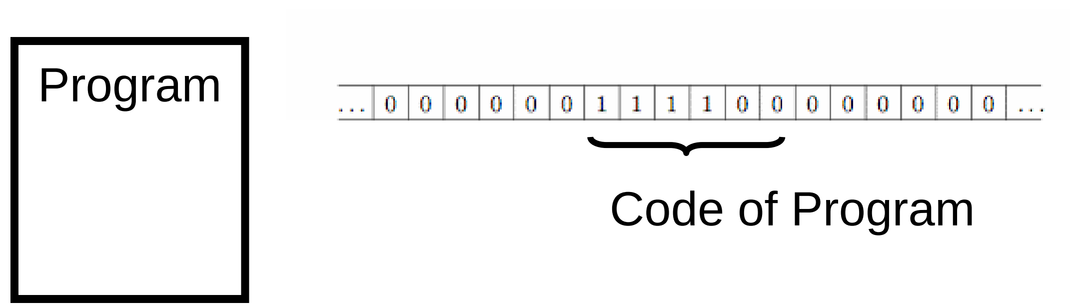
P ⟶ Code (P)

Programs and Data A False Dichotomy

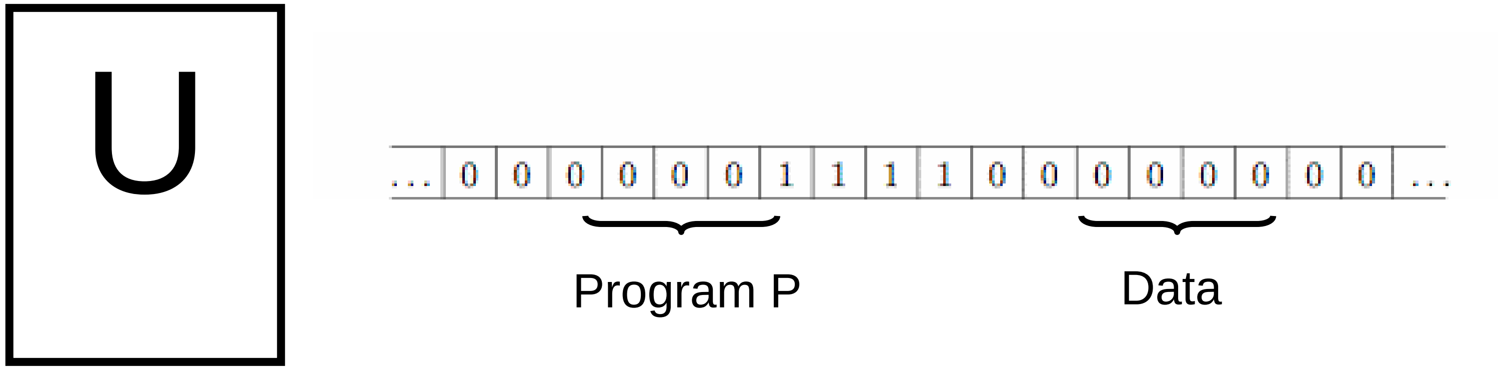
Usual Viewpoint -



But can have -



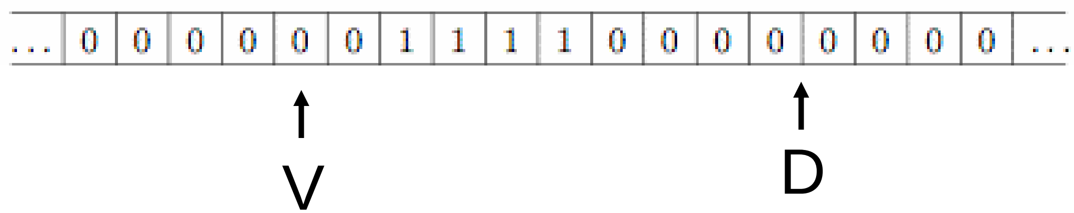
Universal Program U



- 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?

<u>Trivial Idea:</u> Simulate P using universal program U. If P halts will eventually detect

<u>Problem:</u> If P never halts, neither does the simulation.

Next Time: Halting Problem is unsolvable by another program

Also, some class discussion of the two readings.

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