Turing Machines, Busy Beavers, and Big Questions about Computing

David Evans
http://www.cs.virginia.edu/evans

My Research Group
- Computer Security: computing in the presence of adversaries
- Last summer student projects:
  - Privacy in Social Networks (Adrienne Felt)
  - Thwarting Spyware (Meghan Knoll)
  - Hiding Keys in Software (Carly Simpson)

Computer Science = Doing Cool Stuff with Computers?

Toaster Science = Doing Cool Stuff with Toasters?

Computer Science
- Mathematics is about declarative ("what is") knowledge; Computer Science is about imperative ("how to") knowledge
- The Study of Information Processes
  - How to describe them
  - How to predict their properties
  - How to implement them quickly, cheaply, and reliably

Language
Logic
Engineering

Most Science is About Information Processes

Which came first, the chicken or the egg?

How can a (relatively) simple, single cell turn into a chicken?
Understanding Information Processes

- **Art**: How to describe information processes
  - Designing programming languages
  - Inventing algorithms
- **Science**: Predicting properties
  - What resources will a computation consume?
  - Will a program produce the correct output?
- **Engineering**: Implementing processes
  - How to build hardware and software to efficiently carry out information processes

“Computers” before WWII

“Computing is normally done by writing certain symbols on paper. We may suppose this paper is divided into squares like a child’s arithmetic book.”

Alan Turing, *On computable numbers, with an application to the Entscheidungsproblem*, 1936

Mechanical Computing

Modeling Pencil and Paper

“Computing is normally done by writing certain symbols on paper. We may suppose this paper is divided into squares like a child’s arithmetic book.”

Alan Turing, *On computable numbers, with an application to the Entscheidungsproblem*, 1936

Modeling Brains

“For the present I shall only say that the justification lies in the fact that the human memory is necessarily limited.”

Alan Turing

Turing’s Model

- **Input**: 0
  - Write: 1
  - Move: ←
- **Input**: 1
  - Write: 1
  - Move: Halt
What makes a good model?

\[ F = \frac{GM_1M_2}{R^2} \]

Questions about Turing’s Model

- How well does it match “real” computers?
  - Can it do everything they can do?
  - Can they do everything it can do?
- Does it help us understand and reason about computing?

Power of Turing Machine

- Can it add?
- Can it carry out any computation?
- Can it solve any problem?

Universal Machine

Result tape of running \( M \) on Input

A Universal Turing Machine can simulate any Turing Machine running on any Input!

Church-Turing Thesis

- All mechanical computers are equally powerful*  
  *Except for practical limits like memory size, time, energy, etc.
- There exists a Turing machine that can simulate any mechanical computer
- Any computer that is powerful enough to simulate a Turing machine, can simulate any mechanical computer
What This Means

- Your cell phone, watch, iPod, etc. has a processor powerful enough to simulate a Turing machine.
- A Turing machine can simulate the world’s most powerful supercomputer.
- Thus, your cell phone can simulate the world’s most powerful supercomputer (it’ll just take a lot longer and will run out of memory).

Recap

- A computer is something that can carry out well-defined steps.
- All computers are equally powerful.
  - If a machine can simulate any step of another machine, it can simulate the other machine (except for physical limits).
  - What matters is the program that defines the steps.

Are there problems computers can’t solve?

The “Busy Beaver” Game

- Design a Turing Machine that:
  - Uses two symbols (e.g., “0” and “1”)
  - Starts with a tape of all “0”s
  - Eventually halts (can’t run forever)
  - Has \( N \) states
- Goal is to run for as many steps as possible (before halting).

Tibor Radó, 1962

Busy Beaver: \( N = 1 \)

BB(1) = 1 Most steps a 1-state machine that halts can make.

BB(2) = 6
Busy Beaver Numbers

- BB(1) = 1
- BB(2) = 6
- BB(3) = 21
- BB(4) = 107
- BB(5) = Unknown!
  - The best found so far is 47,176,870
- BB(6) > $10^{1730}$

Finding BB(5)

Why not just try all 5-state Turing Machines?
- Transitions from A on 0:
  - 5 possible states + Halt
  - 2 possible write symbols
  - 2 possible directions (L, R)
  - 24 possibilities
- 24 transitions from A on 1
- 5 states: $(24^2)^5$
- $\sim 6.3 \times 10^{13}$

Example

The Halting Problem

- Input: a description of a Turing Machine
- Output: “1” if it eventually halts, “0” if it never halts.

Is it possible to design a Turing Machine that solves the Halting Problem?

Note: the solver must always finish!
Impossible to make Halting Problem Solver

- If it outputs "0" on the input, the input machine would halt (so "0" cannot be correct)
- If it outputs "1" on the input, the input machine never halts (so "1" cannot be correct)

  If it halts, it doesn’t halt!
  If it doesn’t halt, it halts!

Busy Beaver Numbers

- Input: N (number of states)
- Output: BB(N)
  - The maximum number of steps a Turing Machine with N states can take before halting

Is it possible to design a Turing Machine that solves the Busy Beaver Problem?

Summary

- Computer Science is the study of information processes: all about problem solving
  - Almost all science today depends on computing
  - All computers are deeply equivalent
  - Some things cannot be computed by any machine

Challenges

- Specify a number bigger than BB(11111111111111111) on an index card
- Find a TM with 6 states that halts after more than $10^{1730}$ steps
Computer Science at UVa

- New *Interdisciplinary Major in Computer Science* for A&S students
- Take CS150 this Spring
  - Every scientist needs to understand computing, not just as a tool but as a way of thinking
- Lots of opportunities to get involved in research groups

Questions

http://www.cs.virginia.edu/evans
evans@cs.virginia.edu