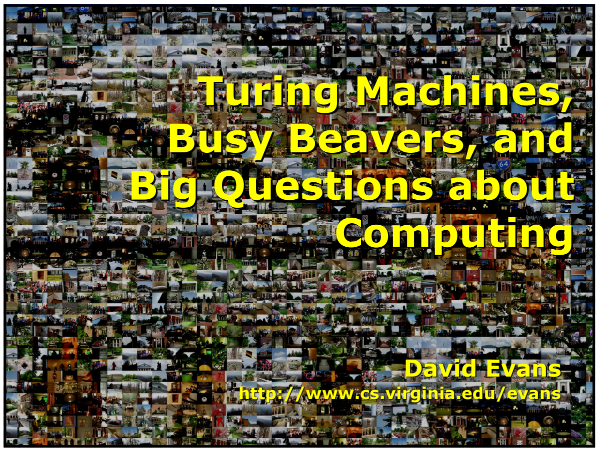


My Research Group

- 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?

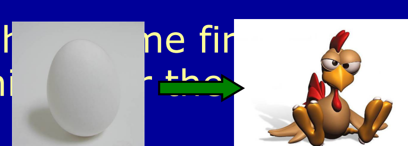
Toaster Science = Doing Cool Stuff with Toasters?

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 — Language
  - How to predict their properties — Logic
  - How to implement them quickly, cheaply, and reliably — Engineering

Most Science is About Information Processes

Wh me fir
chi the



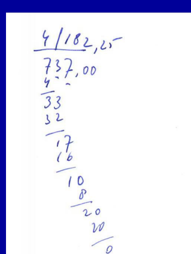
How can a (relatively) simple, single cell turn into a chicken?

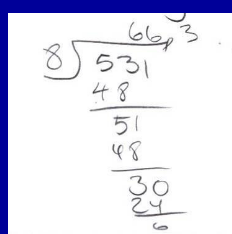
Understanding Information Processes

- 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

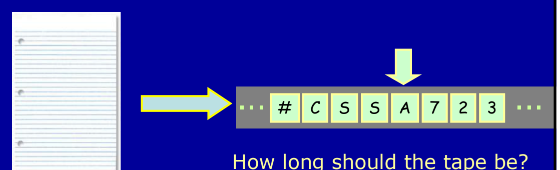
"Computers" before WWII

Mechanical Computing





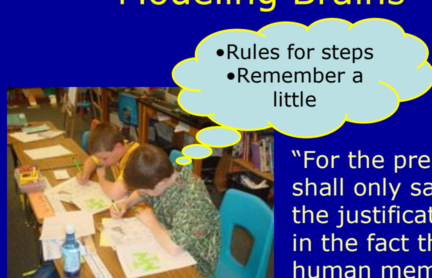
Modeling Pencil and Paper



How long should the tape be?

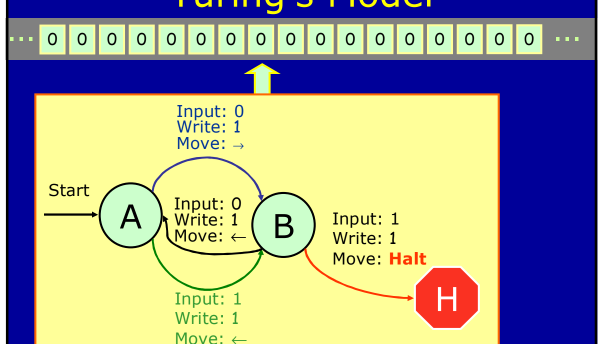
"Computing is normally done by writing certain symbols on paper. We may suppose this paper is divided into squares like a child's arithmetic book."
Alan Turing, *On computable numbers, with an application to the Entscheidungsproblem*, 1936

Modeling Brains

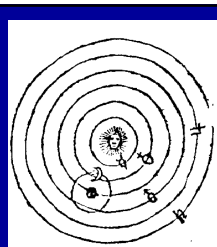


"For the present I shall only say that the justification lies in the fact that the human memory is necessarily limited."
Alan Turing

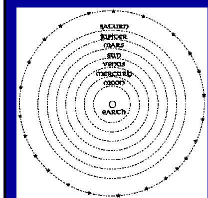
Turing's Model



What makes a good model?



Copernicus



Ptolomy

$F = GM_1M_2 / R^2$

Newton

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

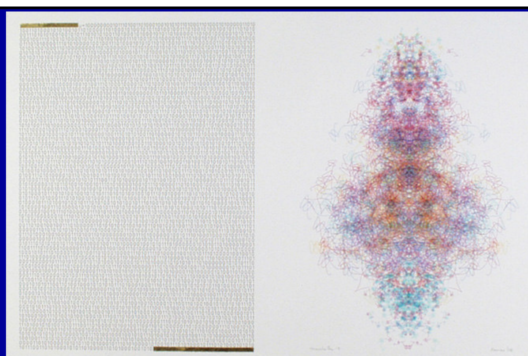
- Can it add?
- Can it carry out any computation?
- Can it solve any problem?

Universal Machine

Result tape of running *M* on Input

Universal Machine

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



Manchester Illuminated Universal Turing Machine, #9
from http://www.verostko.com/manchester/manchester.html

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

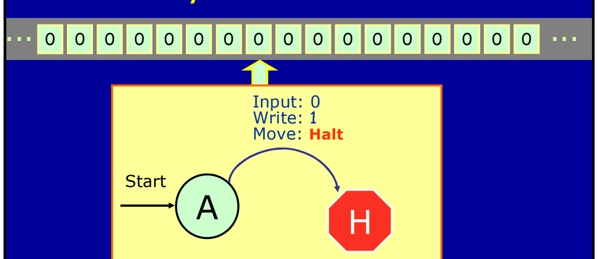
Are there problems computers can't solve?

The "Busy Beaver" Game

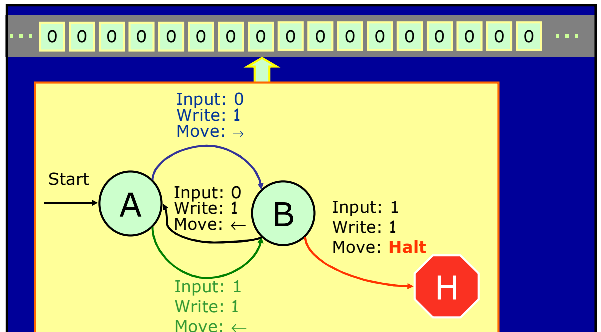
- Design a Turing Machine that:
  - Uses two symbols (e.g., "0" and "1")
  - Starts with a tape of all "0"s
  - Eventually halts (can't run forever)
  - Has *N* states
- Goal is to run for as many steps as possible (before halting)

Tibor Radó, 1962

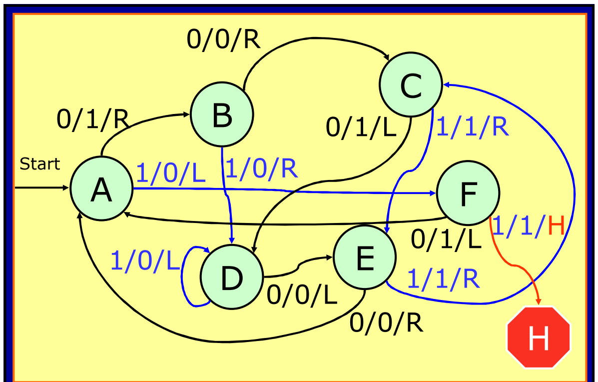
Busy Beaver: N = 1



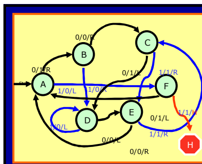
BB(1) = 1 Most steps a 1-state machine that halts can make



BB(2) = 6



6-state machine found by Buntrock and Marxen, 2001



30023277165235628289551030183413401851477543372467525003
73381801735214240760383265881912082978202876698984017860
71345848280422383492822716051848585583668153797251438618
56173020941548768557007853865875730485748722204003076984
40450988713670876150791383110343531646410779192098908371
64477363289374225531955126023251172259034570155087303683
65463087415599082251612993842583069137860727367070819016
05255340770400392265930739979231701547753586298504217125
13378527086223112680677973751790032937578520017666792246
83990885592036293376774476087012844688345547780631649160
18557844268607690279445427980061526931674528213366899174
60886106486574189015401194034857577718253065541632656334
31424232559248670011850671658130342327174896542616040979
71730737166888272814359046394456059281752540483211093060
02474658968108793381912381812336227992839930833085933478
85317657470277606285828915656839229596358626365413938385
67647280513949655544096884565781227432963199608083680945
36421039149584946758006509160985701328997026301708760235
50023959811941059214262166961455282724442921741646549436
38916971139653168926606117092900485806775661787157523545
94049016719278069832866522332923541370293059667996001319
37669855168384885147462515209456711061545198683989449088
56870822449787745514532043585886615939797639351028965232
95803940023673203101744986550732496850436999753711343067
32867615814626929272337566201561282692410545484965841096
15740312114406110889753498991567148886819523660180862466
87712098553077054825367434062671756760070388922117434932
63344477313878371402373589871279027828837719826038006510
50757929252394534506229992082975795848934488862781276290
44163292251815410053522246084552761513383934623129083266
949377380950466643121689746511996847681275076313206

(1730 digits)

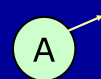
Best found before 2001, only 925 digits!

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: $(24^2)^5$
$\sim 6.3 * 10^{13}$

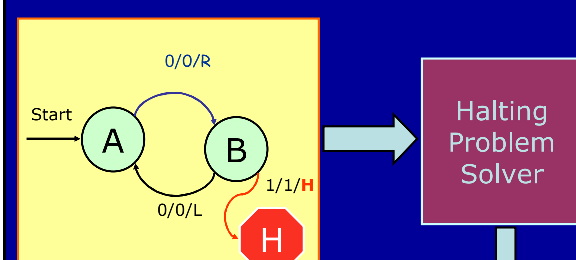
The Halting Problem

- 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?

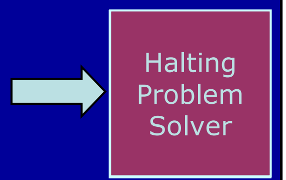
Note: the solver must always finish!

Example

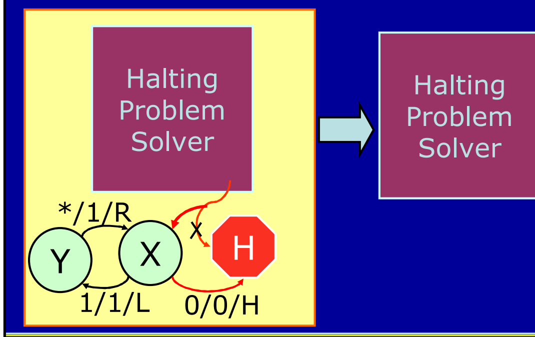


0 (it never halts)

Example



Impossibility Proof!



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(1111111111111111) 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