Announcements/Questions

• Able to locate course web page?
• Able to locate homework and lecture notes?
• Able to see messages on Piazza?
Announcements/Questions

• PDF of book chapters?
• Printed version?
• Solutions for exercises in book
• Open the Room early?
Programming

• Syntax error
• Logic error
• Runtime error

• Debugging

• Read: Soul of the New Machine, Tracy Kidder
A Turing machine is a general example of a CPU that controls all data manipulation done by a computer. More specifically, it is a machine (automaton) capable of enumerating some arbitrary subset of valid strings of an alphabet; these strings are part of a recursively enumerable set.

Assuming a black box, the Turing machine cannot know whether it will eventually enumerate any one specific string of the subset with a given program. This is due to the fact that the halting problem is unsolvable, which has major implications for the theoretical limits of computing.

The Turing machine is capable of processing an unrestricted grammar, which further implies that it is capable of robustly evaluating first-order logic in an infinite number of ways. This is famously demonstrated through lambda calculus.

A Turing machine that is able to simulate any other Turing machine is called a universal Turing machine (UTM, or simply a universal machine). A more mathematically oriented definition with a similar "universal" nature was introduced by Alonzo Church, whose work on lambda calculus intertwined with Turing's in a formal theory of computation known as the Church–Turing thesis. The thesis states that Turing machines indeed capture the informal notion of effective methods in logic and mathematics, and provide a precise definition of an algorithm or "mechanical procedure". Studying their abstract properties yields many insights into computer science and complexity theory.
Turing Machine (simple definition and concept)

• Formalizes and defines concept of a universal computer!
• a tape consisting of slots in which numbers may be written,
• a read/write head that can move along the tape reading the numbers written there or writing numbers onto the tape (or possibly overwriting numbers already written on the tape),
• and a state machine by which it will decide what numbers to write and where on the tape to write them.
Cool thing about Turing Machines!

• It can solve all computable problems!
Church-Turing Thesis

• If you can write an algorithm for performing a computation, then a Turing machine can be built that performs that same computation
Turing Completeness

- A Turing machine is said to be Turing complete if it is capable of simulating any other Turing machine (and therefore of running any algorithm, if you believe Church's Thesis).
- computer
- fixed program
- stored program
- program
- Turing completeness
- interface
- implementation
- abstraction
- ISA (basic)
- programming language (basic)
- compiler (basic)
- microarchitecture (basic)
- architecture (basic)
- registers (basic)
- instruction set (basic)
- syntax (basic)
- semantics (basic)
- library (basic)
- algorithm
- complexity

- trade-off
- low-level language
- high-level language
- syntax error
- logic error
- runtime error
- debugging
- state machine
- microarchitecture
- binary (basic)
- logic gate
- and gate
- truth table
- xnor gate
- Turing machine
- Church-Turing thesis
- Turing complete
- computational complexity
- binary (basic)
Computer

Which Capabilities are being used in each step?

i. Take a number from the user and call it $n_1$.

ii. Take another number from the user and call it $n_2$.

iii. Add $n_1$ and $n_2$. Save the sum in a location. We will call this location $total$.

iv. Divide $total$ by 2. If you get 0 as a remainder, go to step v, else go to step vi.

v. Display “$total$ is even” on the monitor.

vi. Stop

1. Arithmetic
2. Storage
3. Branching
4. Input/Output
1. Start by storing $x/2$ in a storage slot called $s$.
2. Take the current value of $s$ and add $x/s$ to it.
3. Divide $s$ by 2
4. If $s^2 - x$ is bigger than 0.001 or smaller than -0.001, return to step 2.
sqrt

x = 25

y = 1

max_possible = x /2

while (y < max_possible):
    if (y * y > x):
        print "Finished. Sqrt is close to: "
        print y
        break

y = y + 1
Write an algorithm to find the smallest of 3 numbers: n1, n2, and n3?

1. smallest = n1
2. If n2 < smallest

        smallest = n2

3. If n3 < smallest

        smallest = n3
Next Class & Announcements

• HW-1 is due
• Read up through 2.4.1
Logistics
- Not many people want a PDF version of the book
- PDF will be available soon
- Will distribute some solutions
- 3 people volunteer to open door earlier than 10:45

Programming
Errors
- Syntax Errors – error that causes the problem to be unable to compile
- Logic Errors – something logically wrong, like mistyping values
- Runtime Errors – some error occur. E.g, something (disk, network, memory) not available
Debugging – find and fix the problem with the program
Soul of the machine – a book about a team racing to design the next microcomputer starting from the ISA.

Turing Machine
- Formalize/defines the concept of a universal computer
- Capable of reading a tape, writing to a tape, and moving a tape to solve all computable problems

Image above taken from Wikipedia.
Church-Turing thesis: If you can write an algorithm, then a Turing machine can solve it.
A Turing machine is Turning complete if it is capable of simulation any other Turing machine.
Search Engine
For every web page
   Google will visit that webpage at some frequency
   Figure out all the words on that web page
   Store page data in database
Your search terms (keywords) are compared against the database by index
Those webpages that matches the keywords are returned
Google search is divided into the web crawling team, indexing team, networking team, database team, and etc.
   Google wants the change the hardware in the data storage

Terminology Review

In Class Exercise: what capability is each step using
   Capabilities:
      Arithmetic
      Storage
      Branching
      Input/Output
   Steps:
      take number and store it: input and storage
      add and save the sum: arithmetic and storage
      divide by 2: arithmetic
      if 0 then go here, else go there: branching
      display text: input/output

Algorithms Review
   Square Root Algorithms
      Start with $x/2$ and store as $s$, add $s$ to $x/s$, divide by 2, and repeat
      Take any random number, square it, adjust number, and repeat
      Start with counter = 1, square it, increment counter, and repeat until the square is greater
   Find the smallest of 3 numbers
      Scan each number and set as the smallest as you go

Announcements
   HW 1 is due Friday, September 11th
   Read up to Chapter 2.4.1