Computability & Complexity – Why it matters

- Church-Turing thesis
- Halting Problem
- Cardinality of infinite sets
  - Why is it relevant to computer science?
- Undecidable/unsolvable problems

Comments on last Projects

- Bio – Part II
  - We have posted the clusters of the Ito and Gavin data sets (mcl algorithm takes a long time to find them on Ito data)
  - Look at the solution we posted for Part 1

Turing Machine

- Has a tape of infinite length consisting of cells
  - Cells hold 0's or 1's
- At any time, the machine is in one of a finite number of states
- A read/write head accesses one cell of the tape
- One step consists of:
  - read the tape cell, write to it
  - change machine state
  - move head left, right or halt

von Neumann Architecture

- Refers to a computer architecture consisting of
  - a processing unit
  - a single separate storage structure to hold both instructions and data
- Implements a programmable Turing machine
- John von Neumann
  - Austro-Hungarian Mathematician
  - Member of the Manhattan Project

Using Turing machines we can answer the following questions:

Are there problems that have no algorithm?

Yes

Are there functions that are not computable?

Yes

5 Wednesday, April 22, 2009

Church-Turing Thesis

- Church and Turing independently proposed models of computations
  - Church: lambda calculus
  - Turing: Turing machines
- Both capture the notion of computation
- Turing showed that these two very different models were equivalent
- Church was Turing's Ph.D. advisor at Princeton (1936-38)

6 Wednesday, April 22, 2009

Church-Turing Thesis

- Anything we naturally regard as computable is computable by a Turing machine.
- For any algorithm there exists an equivalent Turing machine.
- A Turing machine can do anything that can be described by a mechanical process.

In order to study computation, we only need to study Turing machines.

If a process cannot carried out by a Turing machine, then it cannot be computed.

7

Wednesday, April 22, 2009

Are there problems for which there exists no algorithm?

YES, many.

They are called unsolvable/not computable problems.

Halting Problem (Zelle 13.4.2)

The halting problem is the most famous of all unsolvable problems.

8

Wednesday, April 22, 2009

Write a program HALT to solve the following

Input for program HALT

code of an arbitrary program P and an input I

Task

determine whether program P halts on input I

Output of program HALT

1 if P halts on input I

0 if P does not halt in input I

Program HALT detects whether program P has an infinite loop.

HALT solves the Halting Problem

Claim: Program HALT cannot exist

The Halting problem is not a computable problem.

We also say it is not a decidable problem.

If the Halting problem could be solved, it would solve a number of conjectures.

Goldbach's Conjecture

Every even number greater than 2 is the sum of two primes.

In 1742, Goldbach sent the conjecture to Euler who found it interesting, but could not prove it.

If the Halting Problem can be solved, Goldbach's conjecture would be decided.

```
Program P (has no input)
    n =2
    while True:
        if 2*n is not the sum of two prime numbers, then HALT
        n = n+1
```


Which problem should you agree to write an algorithm for?

Problem 1

Does there exist a positive integer-valued solution to

$$313(a^3 + b^3) = c^3$$

Problem 2

Given a positive integer, reverse the digits and add it to the original number. Repeat until you get a palindrome.

5280 + 0825 = 6105

6105 + 5016 = 11121

11121 + 12111 = **23232**

A. Problem 1
B. Problem 2
C. Neither
D. Both

Should be C)

Problem 1: Does there exist a positive integer-valued solution to $313(a^3 + b^3) = c^3$?

Answer: yes (has about 1000 digits)

(*Fermat's Last Theorem:* $a^n + b^n = c^n$ has no solution for $n>2$)

Problem 2: Given a positive integer, reverse the digits and add it to the original number. Repeat until you get a palindrome.

Answer: not know to terminate for 196 (first 9 million iterations do not)

13 Wednesday, April 22, 2009

Which infinite set is not like the others?

A. N = {0, 1, 2, 3, 4, 5, 6, 7, … }
B. Z = { … , -3, -2, -1, 0, 1, 2, 3, 4, … }
C. Q = set of rational numbers
D. R = set of all real numbers
E. E = {0, 2, 4, 6, 8, 10, … }

14 Wednesday, April 22, 2009

Cardinality of infinite sets

Two sets A and B have the same *cardinality* if and only if there is a 1-1 correspondence from A to B.

- Consider sets E and N
  E is a proper subset of N.
  Yet, E and N have the same cardinality: f(x) = 2x

0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**…**
0	**2**	**4**	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**	

15 Wednesday, April 22, 2009

Infinite sets of the same cardinality

- Consider sets N and Z

f(x) = x/2 if x is even
- (x+1)/2 if x is add

0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**…**
0	**-1**	**1**	**-2**	**2**	**-3**	**3**	**-4**	**4**	**-5**	**5**	

16 Wednesday, April 22, 2009

Infinite sets of the same cardinality

Consider sets N and Q (positive integers and rational numbers)

- Surprisingly, there exists a 1-1 mapping
- Functions are harder/messier to express

1/1 1	1/2 2	1/3 4	1/4 7	1/5 11	1/6 16	1/7 22	1/8 29	..
2/1 3	2/2 5	2/5 8	2/4 12	2/5 17	2/6 23	2/7 30	2/8	..
3/1 6	3/2 9	3/3 13	3/4 18	3/5 24	3/6 31	3/7	3/8	..
4/1 10	4/2 14	4/3 19	4/4 25	4/5 32	4/6	4/7	4/8	..
5/1 15	5/2 20	5/3 26	5/4 33	5/5	5/6	5/7	5/8	..
6/1 21	6/2 27	6/3 34	6/4	6/5	6/6	6/7	6/8	..
7/1 28	7/2 35	7/3	7/4	7/5	7/6	7/7		..
8/1 36	8/2	8/3	8/4	8/5	8/6			..
.	.							

Countable sets

A set is *countable* if

- It is finite, or
- It has the same cardinality as the natural numbers N

Countable sets: N, E, Z, Q

Uncountable set: real numbers R

The real numbers between 0 and 1 are not countable

Assume they are countable. Then they can be listed as:

0.	d1,1	d1,2	d1,3	d1,4	d1,5	d1,6	d1,7	d1,8	…
0.	d2,1	d2,2	d2,3	d2,4	d2,5	d2,6	d2,7	d2,8	…
0.	d3,1	d3,2	d3,3	d3,4	d3,5	d3,6	d3,7	…	…
0.	d4,1	d4,2	d4,3	d4,4	d4,5	d4,6	d4,7	…	
0.	d5,1	d5,2	d5,3	d5,4	d5,5	…			
…									
…									

Change the digits in the diagonal:

If it is a 4, set it to 5. Otherwise set it to 4.

0.	4	d1,2	d1,3	d1,4	d1,5	d1,6	d1,7	d1,8	…
0.	d2,1	5	d2,3	d2,4	d2,5	d2,6	d2,7	d2,8	…
0.	d3,1	d3,2	5	d3,4	d3,5	d3,6	d3,7		…
0.	d4,1	d4,2	d4,3	4	d4,5	d4,6	d4,7		
0.	d5,1	d5,2	d5,3	d5,4	4				
0.						4			
.							5		

Let $r = 0.r_1r_2r_3r_4r_5 \ldots$ be the number generated by the new diagonal values (0.4554445 …)

Real number r cannot be in this list! Why?

A digit on the diagonal cannot be 4 and 5 at the same time.

21

Putting it all together

- The assumed listing/enumeration of all real numbers between 0 and 1 cannot exist.
- **This means set R is uncountable.**

The proof technique we used is called **diagonalization.**

- it negates/flips the values on the diagonal to obtain a contradiction

22

Wednesday, April 22, 2009

Assume the Halting problem is decidable.

Then there exists a program HALT that given any program P and an input I decides whether P halts on I (returning a 1 means P halts, 0 means not halting)

NOT COVERED IN CLASS

Create a new program **STRANGE** which takes as an input any program P:

1. STRANGE calls HALT with P and P as inputs
2. **if** HALT(P,P) returns 0 (i.e., P on P does not halt)
   then STRANGE halts
   else STRANGE goes into an infinite loop

What does STRANGE do on input STRANGE?

It calls HALT with STRANGE and STRANGE as inputs:

if HALT(STRANGE, STRANGE) returns 0 (i.e., STRANGE on input STRANGE does not halt)
then STRANGE halts
else STRANGE goes into an infinite loop

NOT COVERED IN CLASS

But, STRANGE cannot halt and not halt at the same time!

Hence, program HALT cannot exist

Implications

- There exists no program solving the Halting problem
- There exists an infinite number of unsolvable problems
  - There exist more unsolvable problems than solvable ones
  - Fortunately, most problems we need to solve are solvable
- Being solvable does not mean not mean practically solvable
- Most optimization problems arising in applications take too much time to be solved exactly