CSE 2500

Introduction to Discrete Systems

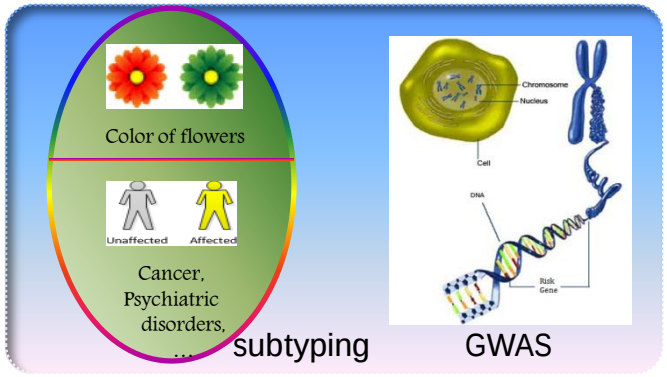
Jinbo Bi

Department of Computer Science & Engineering

http://www.engr.uconn.edu/~jinbo

The Instructor

- Ph.D. in Mathematics
- Working experience
  - Siemens Medical Solutions
  - Department of Defense, Bioinformatics
  - UConn, CSE
- Contact: jinbo@engr.uconn.edu, 486-1458 (office)
- Research Interests:
  - Genetic analysis of complex human disease
  - Intelligent online counseling systems to monitor and regulate heavy alcohol use among college students
  - Clinical decision support systems – Easy Breathing for asthma care



subtyping GWAS

http://labhealthinfo.uconn.edu/EasyBreathing

Today

- Organizational details
- Purpose of the course
- Material coverage
- Section 1.1 of the text

Course Syllabus

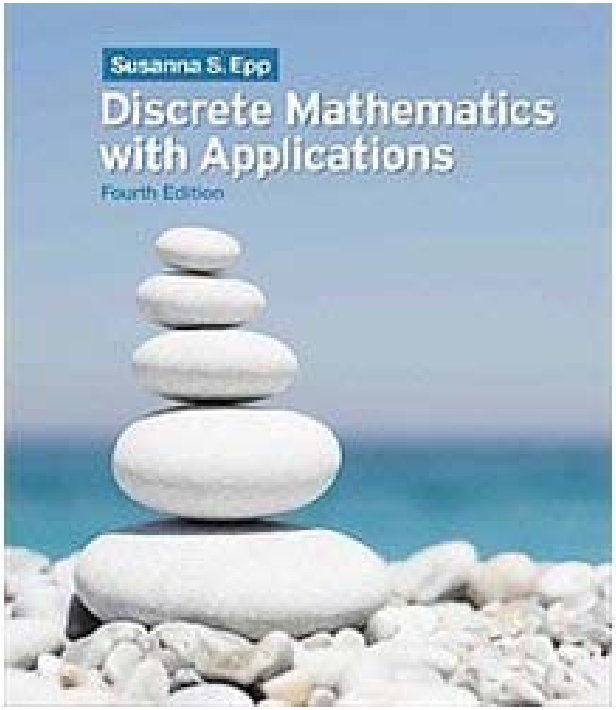
- Go over syllabus carefully, and keep a copy of it

- Course website

  http://www.engr.uconn.edu/~jinbo/Fall2012_discrete_math.htm

Textbook

- Attending the lectures is **not** a substitute for reading the text
- Lectures merely highlight some examples
- Read the text and do as many exercises as humanly possible



Slides

- We may not have slides for each lecture (only the first lecture has a full set of slides)
- If a lecture uses slides, they will be available right after lecture at HuskyCT

Marking Scheme

- 6 out of 7 assignments 30%
  - (the lowest mark assignment will be dropped)
- 2 Midterm: 40%
- 1 Final Exam: 30%

- Curved
  - Curve is tuned to the final overall distribution
  - No pre-set passing percentage

In-Class Participation

- Finding errors in my lecture notes
- Answering my questions
- Asking questions on the material

Assignments

- 6 out of 7 will count
- Each will have 4-10 problems from the textbook
- Solutions will be published at HuskCT after due day
- Each assignment will be given 1 week, but some may be 2-weeks

Some Arithmetic

- Suppose you are running late and the due date is near...
- If you don't hand in just one assignment:
  - No problem (only 6 out of 7 count)
- If you don't hand in two assignments:
  - Lose 30 / 6 = 5% of the final mark
- Suppose you copy it from a friend:
  - We will most likely detect it
  - You will lose 100% of the final mark
  - Your friend will be prosecuted as well
  - Plus: suspension, possible deportation, etc.
- The choice is always yours...

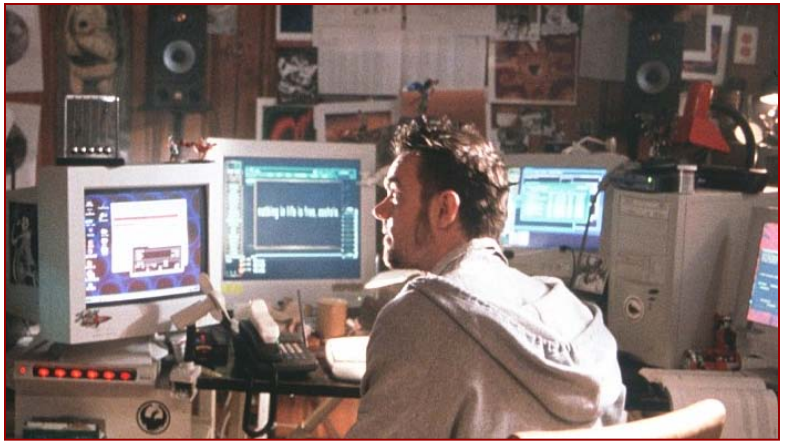
Questions?

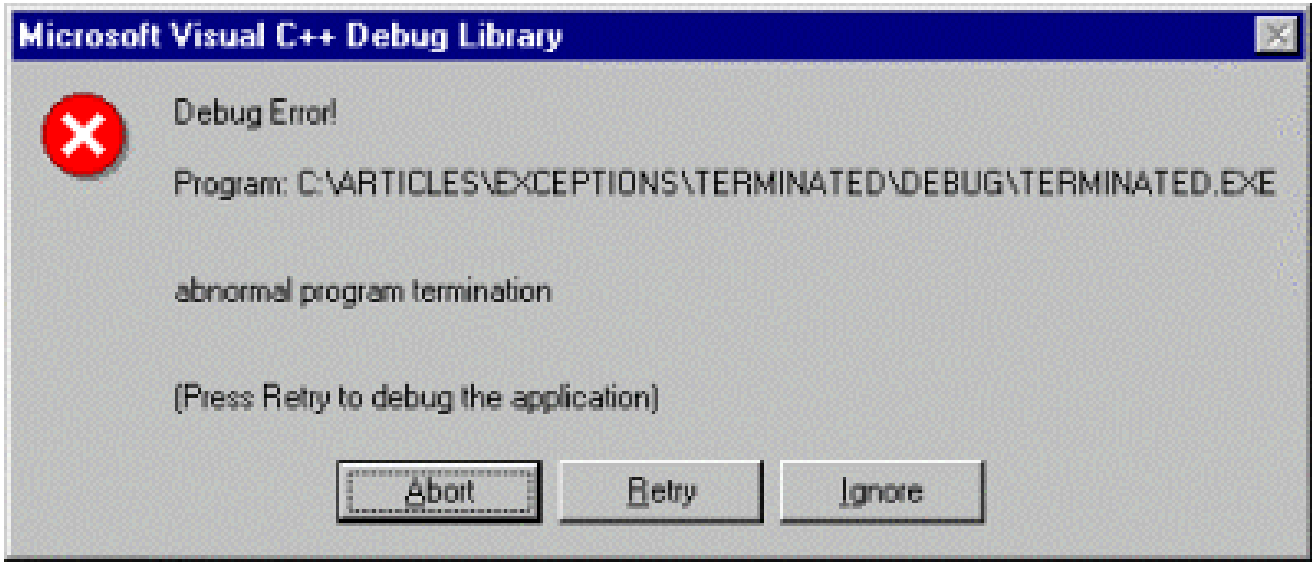
Why This Course?

- Relation to real life:
  - Algorithm correctness ~ programming, reverse-engineering, debugging
  - Propositional logic ~ hardware (including VLSI) design
  - Sets/relations ~ databases (Oracle, MS Access, etc.)
  - Predicate logic ~ Artificial Intelligence, compilers
  - Proofs ~ Artificial Intelligence, VLSI, compilers, theoretical physics/chemistry

Code Correctness

- Millions of programmers code away daily...
- How do we know if their code works?
- How can we find the bug if it does not work

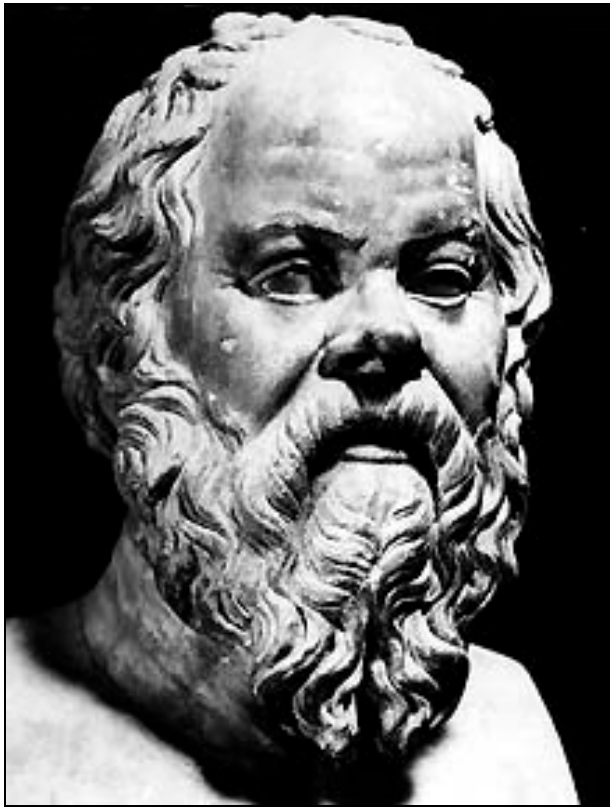




Argument #1

- All men are mortal
- Socrates is a man

- Therefore,
  Socrates is mortal



Argument #2

- Nothing is better than God
- A sandwich is better than nothing

- Therefore,
  a sandwich is better
  than God

Validity

- An argument is valid if and only if given that its premises hold its conclusion also holds

- So…
  - Socrates argument: Valid or Invalid?
  - Sandwich argument: Valid or Invalid?

How can we tell ?

- Common sense?
- Voting?
- Authority?
- What is valid argument anyway?
- Who cares?

- ???

CSE 2500

- Logic : a formal way to assess a validity of an argument
- Can prove theorems in a semi-automatic fashion
- Can verify given proofs that an argument is valid

Material Coverage

- Chapter 1: speaking mathematically
- Chapter 2: The logic of compound statements
- Chapter 3: The logic of quantified statements
- Chapter 4: elementary number theory
- Chapter 5: mathematical induction, Recursion
- Chapter 6: set theory
- Chapter 7: functions
- Chapter 8: Relation
- Chapter 10: Graphs (potential)

Questions?

Arguments in Puzzles

- The Island of Knights and Knaves

Never lie

Always lie

Example #1

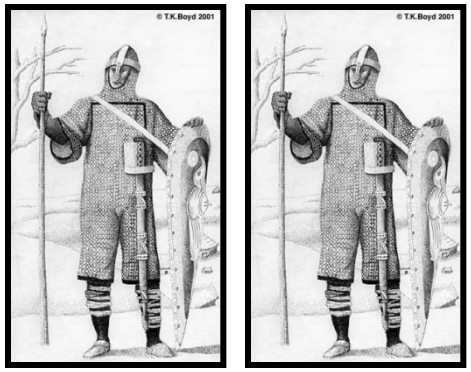
- You meet two people: A, B
- A says:
  - I am a Knave or
  - B is a Knight.

- Who is A?

- Who is B?

Solution

- The original statement can be written as:
- S = X or Y
- X = “A is a Knave”
- Y = “B is a Knight”
- Suppose A is a Knave
- Then S must be false since A said it
- Then both X and Y are false
- If X is false then A is not a Knave
- Contradiction : A cannot be a Knave and not a Knave !
- So A must be a Knight
- So S is true and X is not true
- Thus, to keep S true Y must be true
- So B is a Knight too



How about…

- You meet just one guy : A
- A says:
  - “I’m a Knave!”
- Who is A?

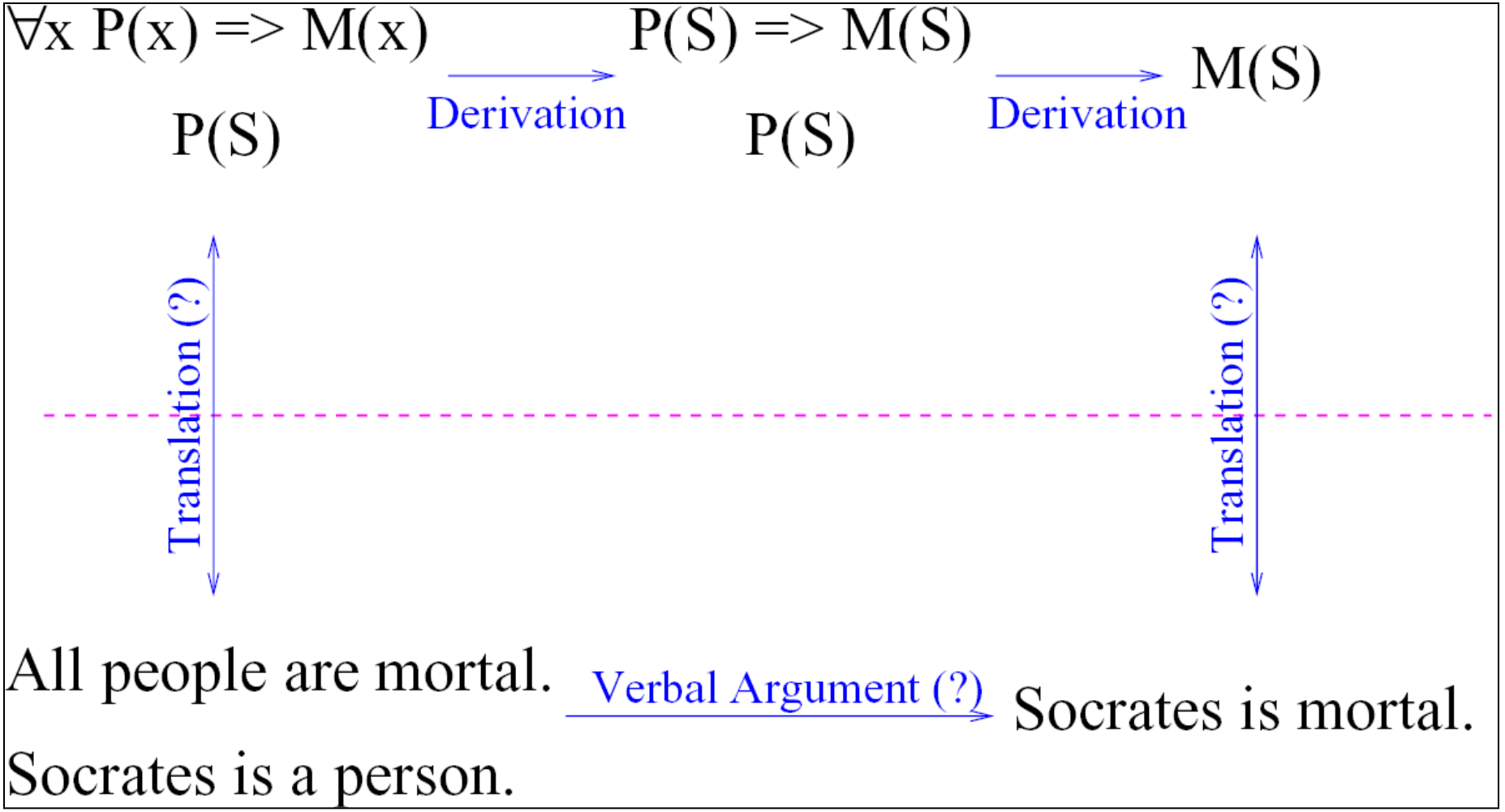
Features Of An Argument

- arguments involve things or objects
- things have properties
- arguments consist of statements
- statements may be composed
- an argument starts with assumptions which create a context.
- each step yields another statement which is true, within its context.
- arguments may contain sub-arguments
- it is absurd for a statement to be both true and false

Formalization

- *Why formalize?*
  - to remove ambiguity
  - to represent facts on a computer and use it for proving, proof-checking, etc.
  - to detect unsound reasoning in arguments

Graphically…



Logic

- Mathematical logic is a tool for dealing with formal reasoning
- Logic does:
  - Assess if an argument is valid/invalid
- Logic does not directly:
  - Assess the truth of atomic statements

Differences

- Logic can deduce that:
  - Edmonton is in Canada
- given these facts:
  - Edmonton is in Alberta
  - Alberta is a part of Canada
- and the definitions of:
  - ‘to be a part of’
  - ‘to be in’

- Logic knows nothing of whether these facts actually hold in real life!

Questions?

Mathematical Symbols

- Simplest kind of math logic

- Dealing with:
  - **Propositions**: X,P,Q,...
    each can be true or false
    Examples: P="*I'm a knave*"
    Q=*"He is a knight"*
  - **Connectives**: &, v, →, ⇔, ~, ...
    - connect propositions: X v Y

Connectives

- Different notation is in use
- We will use the common math notation:
  - ~ not
  - ∨ or (non-exclusive!)
  - ∧ and
  - → implies (if ... then ...)
  - ⇔ if and only if
  - ∀ for all
  - ∃ there exists

Formulae

- A statement/proposition: true or false
- Atomic: P, Q, X, Y, ...
- Unit Formula: P, ~P, (formula), ...
- Conjunctive: P & Q, P & ~Q, ...
- Disjunctive: P v Q, P v (P & X),...
- Conditional: P → Q
- Biconditional: P ⇔ Q

Determining Truth of A Formula

- Atomic formulae: given
- Compound formulae: via meaning of the connectives
- Suppose: P is true
  Q is false
  How about: (P v Q)
- Truth tables

Truth Tables

- Suppose: P is false
  Q is false
  X is true

- How about:
  - P & Q & X
  - P v Q & X
  - P & Q v X

Precedence

- ~ *highest*
- &
- v
- →, ⇔ *lowest*

- Avoid confusion - use ‘(‘ and ‘)’:
  - P & Q v X
  - (P & Q) v X

Parenthesizing

- Parenthesize & build truth tables
- Similar to arithmetics:
  - 3*5+7 = (3*5)+7 but **NOT** 3*(5+7)
  - A&B v C = (A&B) v C but **NOT** A&(B v C)

- So start with sub-formulae with highest-precedence connectives and work your way out

- Let's do the knave & knight problem in TT

TT for K&K

- S = X or Y
- X = “A is a Knave”
- Y = “B is a Knight”

A	B	S	X	Y	X v Y	Absurd
Knave	Knave	false	true	false	true	yes
Knave	Knight	false	true	true	true	yes
Knight	Knave	true	false	false	false	yes
Knight	Knight	true	false	true	true	no

Questions?