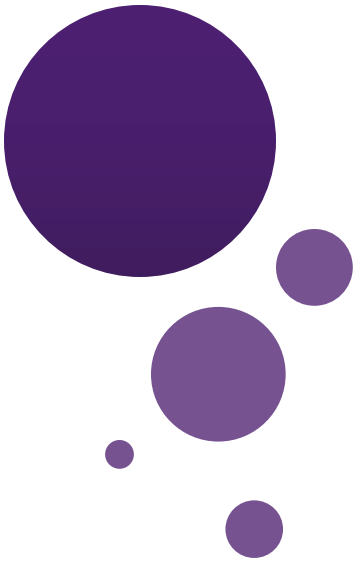



# Lecture 1: Introduction, Logic and Proofs

Dr. Chengjiang Long
Computer Vision Researcher at Kitware Inc.
Adjunct Professor at SUNY at Albany.
Email: **clong2@albany.edu**

# Self-introduction

2009.6 B.S. Wuhan, China

2011.6 M.S. Wuhan, China

2013.5-8 Research Intern, NEC Laboratories America, Cupertion, CA

2015.5-8 Research Intern, GE Global Research, Niskayuna, NY

2015.10 Ph.D. Stevens Institute of Technology, Hoboken, NJ

2016.2-now Computer Vision Researcher, Senior R&D Engineer, Kitware, Clifton Park, NY

2018.1-5 Adjunct Professor, Rensselaer, Troy, NY

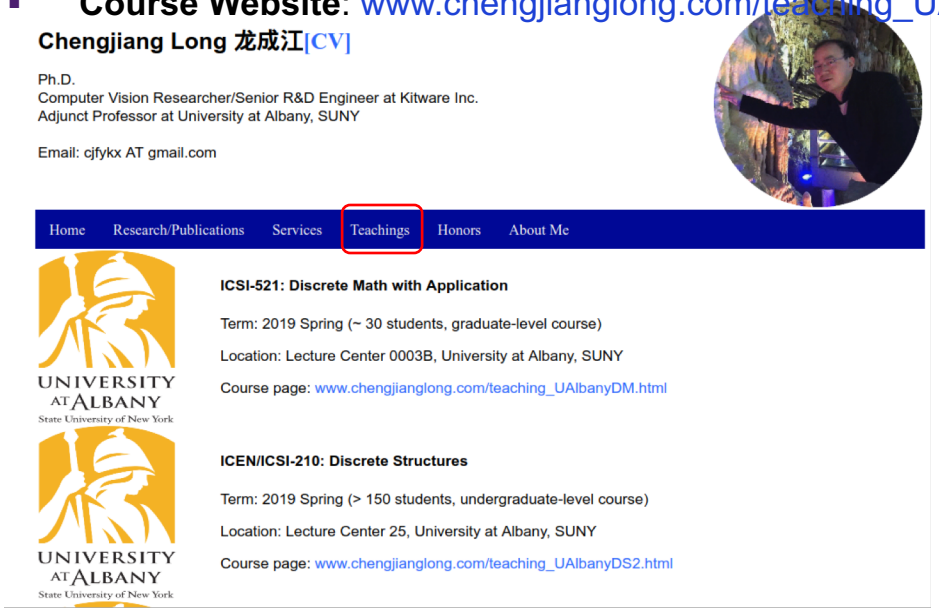
2018.8-now Adjunct Professor, University at Albany, State University of New York, Albany, NY





# Course information

- **ISCI-521** Discrete Math with Application
- **Term**: Spring 2019
- **Instructor**: Dr. Chengjiang Long
- **Email:** clong2@albany.edu
- **Class time**: 2:45 pm—5:35 pm, Monday, Wednesday & Friday
- **Location**: LC 0003B
- **Teaching Assistant**: Oguz Aranay (oaranay@albany.edu) and Tuan N Tran (ttran3@albany.edu)
- **Course Website**: www.chengjianglong.com/teaching_UAlbanyDM.html



**Chengjiang Long 龙成江[CV]**

Ph.D.
Computer Vision Researcher/Senior R&D Engineer at Kitware Inc.
Adjunct Professor at University at Albany, SUNY

Email: cjfykx AT gmail.com

Home | Research/Publications | Services | Teachings | Honors | About Me

**ICSI-521: Discrete Math with Application**

Term: 2019 Spring (~ 30 students, graduate-level course)

Location: Lecture Center 0003B, University at Albany, SUNY

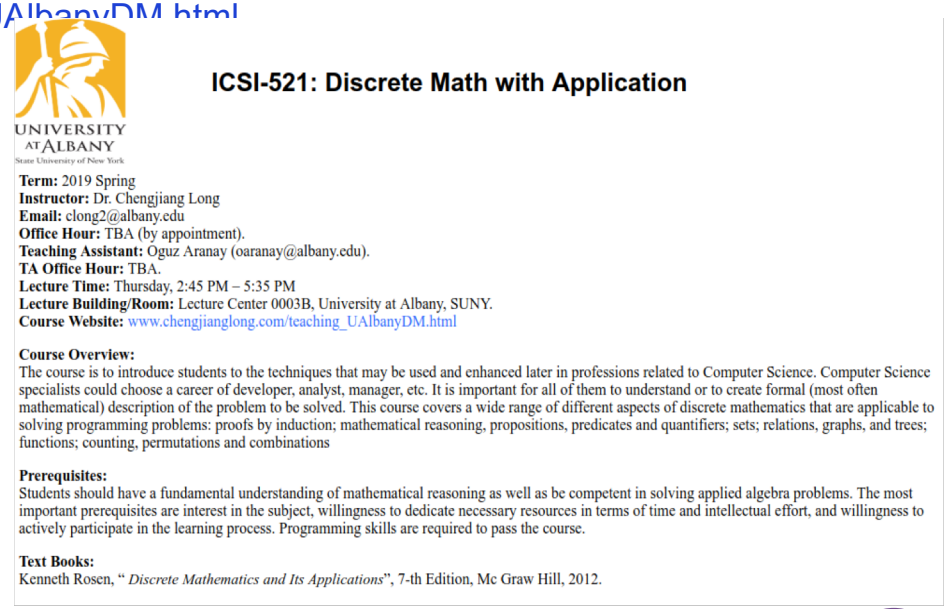
Course page: www.chengjianglong.com/teaching_UAlbanyDM.html

**ICEN/ICSI-210: Discrete Structures**

Term: 2019 Spring (> 150 students, undergraduate-level course)

Location: Lecture Center 25, University at Albany, SUNY

Course page: www.chengjianglong.com/teaching_UAlbanyDS2.html



**ICSI-521: Discrete Math with Application**

**Term:** 2019 Spring
**Instructor:** Dr. Chengjiang Long
**Email:** clong2@albany.edu
**Office Hour:** TBA (by appointment).
**Teaching Assistant:** Oguz Aranay (oaranay@albany.edu).
**TA Office Hour:** TBA.
**Lecture Time:** Thursday, 2:45 PM – 5:35 PM
**Lecture Building/Room:** Lecture Center 0003B, University at Albany, SUNY.
**Course Website:** www.chengjianglong.com/teaching_UAlbanyDM.html

**Course Overview:**
The course is to introduce students to the techniques that may be used and enhanced later in professions related to Computer Science. Computer Science specialists could choose a career of developer, analyst, manager, etc. It is important for all of them to understand or to create formal (most often mathematical) description of the problem to be solved. This course covers a wide range of different aspects of discrete mathematics that are applicable to solving programming problems: proofs by induction; mathematical reasoning, propositions, predicates and quantifiers; sets; relations, graphs, and trees; functions; counting, permutations and combinations

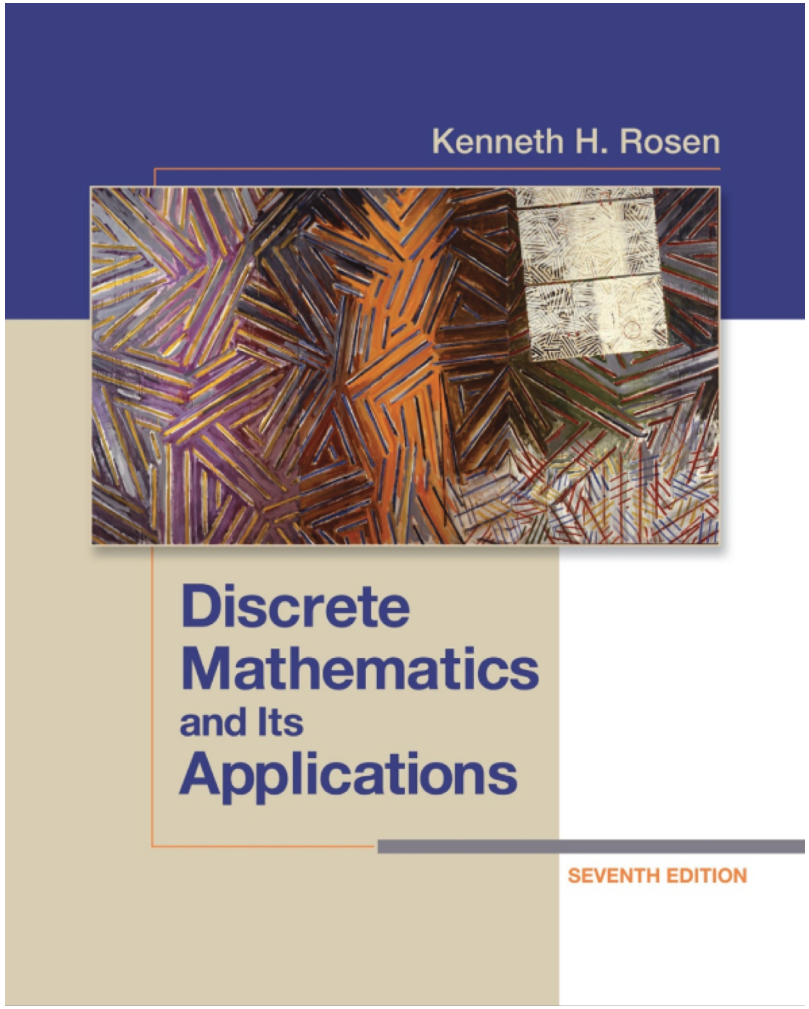
**Prerequisites:**
Students should have a fundamental understanding of mathematical reasoning as well as be competent in solving applied algebra problems. The most important prerequisites are interest in the subject, willingness to dedicate necessary resources in terms of time and intellectual effort, and willingness to actively participate in the learning process. Programming skills are required to pass the course.

**Text Books:**
Kenneth Rosen, "*Discrete Mathematics and Its Applications*", 7-th Edition, Mc Graw Hill, 2012.

# Topics and textbooks

Topics covering:

- Logic
- Proof
- Sets
- Functions
- Counting
- Discrete probability
- Relations
- Graph
- Tree




This textbook has been used in over 500 institutions around the world.

*http://www2.fiit.stuba.sk/~kvasnicka/Mathematics%20for%20Informatics/Rosen_Discrete_Mathematics_and_Its_Applications_7th_Edition.pdf*

# Prerequisites

- Students should have a fundamental understanding of mathematical reasoning as well as be competent in solving applied algebra problems.
- The most important prerequisites are interest in the subject, willingness to dedicate necessary resources in terms of time and intellectual effort, and willingness to actively participate in the learning process.
- Programming skills are required to pass the course!

# Objectives of This Course

- The goal of the course is to introduce students to the techniques that may be used and enhanced later in professions related to Computer Science.
  - To learn basic mathematical concepts, e.g. sets, functions, graphs
  - To be familiar with formal mathematical reasoning, e.g. logic, proofs
  - To improve problem solving skills
  - To see the connections between discrete mathematics and computer science

# Grading

- **Class participation: 5%**
- **5 Homework assignments: 50%**
- **1 Midterm exams: 15%**
- **Final project: 30%**
- **Final grade:**
  - ① **A(≥92), A-(≥90),**
  - ② **B+(≥87), B(≥82), B-(≥80),**
  - ③ **C+(≥77), C(≥72), C-(≥70),**
  - ④ **D+(≥67), D(≥62), D-(≥60),**
  - ⑤ **F(< 60).**

# Optional Problems and Attendance Bonus

- Extra points: 20% for each homework. Note that this is optional, the purposes of this design is to encourage the self-motivated students to challenge themselves and give them more chances to get a higher score.

- Attendance bonus: I would like to give the bonus to reward those students whose attendance is less than 3. For those who never miss any class, I will give them 5 extra points on the final grade. For those who miss only 1 class, I will give them 2 extra points. And for those who miss 2 classes, I will give them 1 extra point on the final grade.

# Schedule

| Class | Date | Topic | Reading | Homework | Slides |
|---|---|---|---|---|---|
| 1 | 1/24/2019 | Introduction, Logic and Proofs | Ch 1 | | Lecture_1 |
| 2 | 1/31/2019 | Induction I and II | Ch 5 | Homework_1 | Lecture_2 |
| 3 | 2/7/2019 | Basic Structures: Sets, Function, Sequences etc. | Ch 2 | | Lecture_3 |
| 4 | 2/14/2019 | Algorithm, Growth Function and Complexity | Ch 3 | Homework_2 | Lecture_4 |
| 5 | 2/21/2019 | Number Theory | Ch 4 | | Lecture_5 |
| 6 | 2/28/2019 | Basics of Counting | Ch 6 | Homework_3 | Lecture_6 |
| 7 | 3/7/2019 | Probability and Applications | Ch 7 | | Lecture_7 |
| 8 | 3/14/2019 | Advanced Counting Techniques (1) | Ch 8 | Homework_4 | Lecture_8 |
| | 3/21/2019 | Class Suspended -- Spring Break | | | |
| 9 | 3/28/2019 | Advanced Counting Techniques (2) | Ch 8 | Homework_5 | Lecture_10 |
| 10 | 4/4/2019 | Relations | Ch 9 | | Lecture_11 |
| 11 | 4/11/2019 | Midterm Exam | Ch 1-9 | Midterm_Exam | |
| 12 | 4/18/2019 | Graph Thoery | Ch 10 | | Lecture_12 |
| 13 | 4/25/2019 | Trees | Ch 11 | | Lecture_13 |
| 14 | 5/2/2019 | Final Project Presentation | | | |

Note: The above course schedule may be subject to change. Please do check the latest update.

# Rules

- **Need to be absent from class?**
  - 1 point per class: please send notification and justification at least 2 days before the class.
- **Late submission of homework?**
  - The maximum grade you can get from your late homework decreases 25% per day.
  - **> 3 hours after the deadline** will be considered as "late submission".
- **Zero tolerance on plagiarism**!!!
  - The first time you receive zero grade for the assignment.
  - The second time you get "F" in your final grade.
  - Refer to the University at Albany, SUNY's honor system for your behavior.

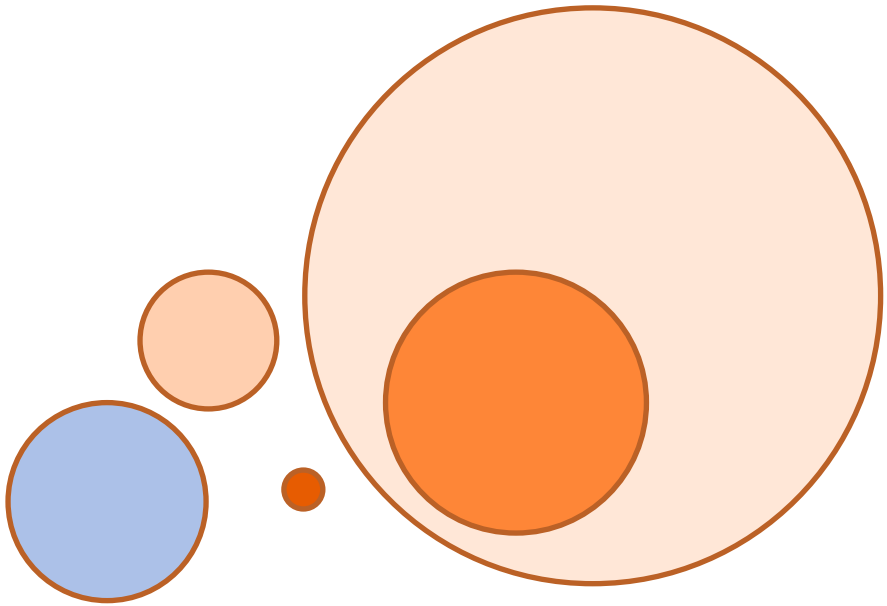
# Why Mathematics?

- Mathematics is the tool that arms practitioners with form (theoretical) approaches to problem solving.

- Formal approach allows:
  - to manage very small and very large numbers.
  - to resue solutions.

*Example:*

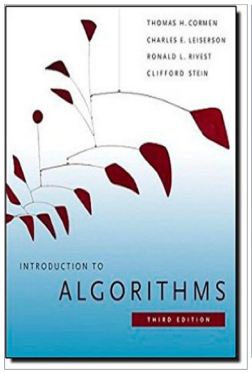
Area of a circle $A = \pi \times r^2$

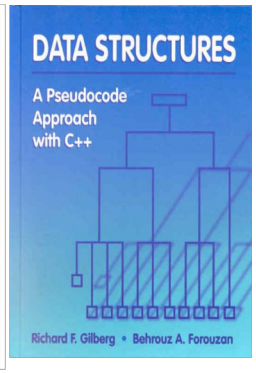


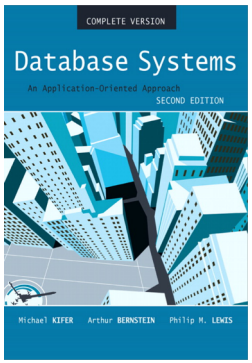
# Why Mathematics?

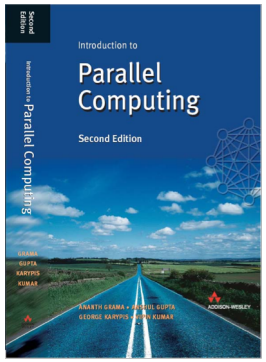
Design efficient computer systems.

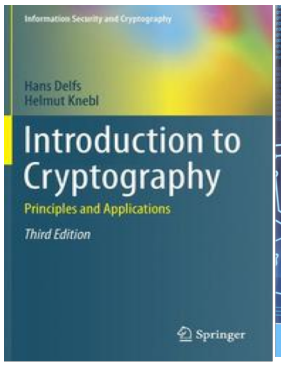
- How did Google manage to build a fast search engine?
- What is the foundation of internet security?

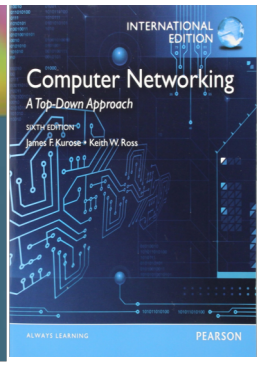













...

*algorithms, data structures, database, parallel computing, distributed systems, cryptography, computer networks…*

Logic, number theory, counting, graph theory…

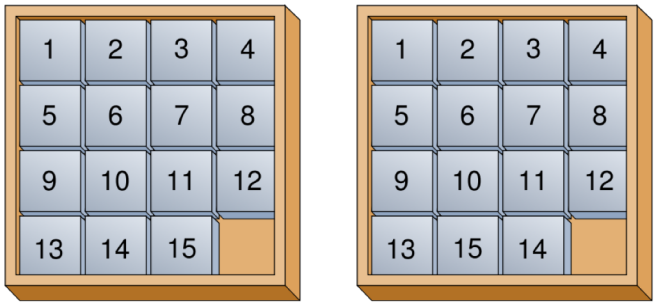
# Topic 1: Logic and Proofs

How do computers think?

Logic: propositional logic, first order logic

Proof: induction, contradiction

$$\frac{x_1 + x_2 + \ldots + x_n}{n} \geq \sqrt[n]{x1 \cdot x_2 \cdots x_n}$$




Artificial intelligence, database, circuit, algorithms

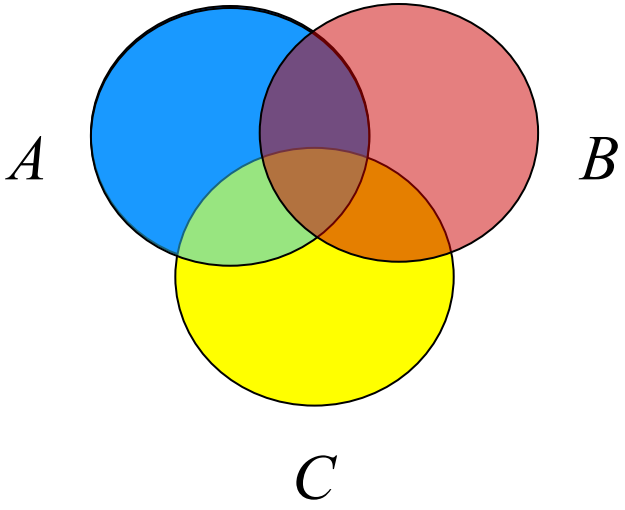
# Topic 2: Number Theory

- Number sequence
- (Extended) Euclidean algorithm
- Prime number, modular arithmetic, Chinese remainder theorem
- Cryptography, RSA protocol

Cryptography, coding theory, data structures

# Topic 3: Counting

- Sets and Functions
- Combinations, Permutations, Binomial theorem
- Counting by mapping, pigeonhole principle
- Recursions




Probability, algorithms, data structures

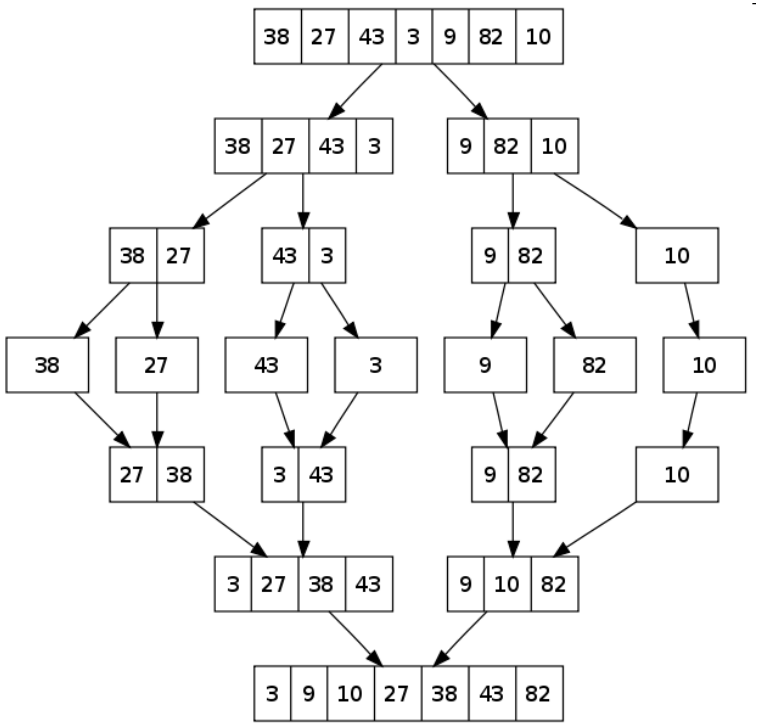
# Topic 3: Counting

How many steps are needed to sort n numbers?

**Algorithm 1 (Bubble Sort):**

Every iteration moves the i-th smallest number to the i-th position

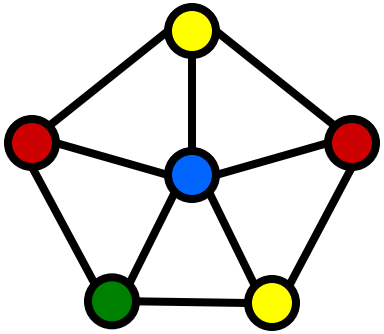
**Algorithm 2 (Merge Sort):**

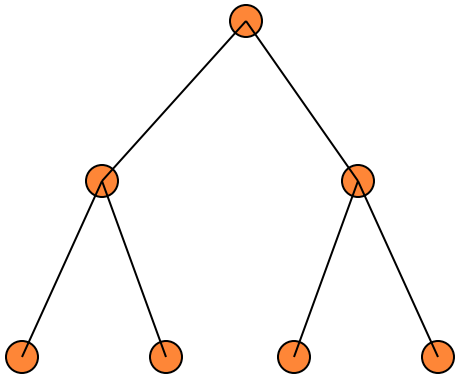



Which algorithm runs faster?

# Topic 4: Graph Theory

- Graphs, Relations
- Degree sequence, Eulerian graphs, isomorphism
- Trees
- Matching
- Coloring

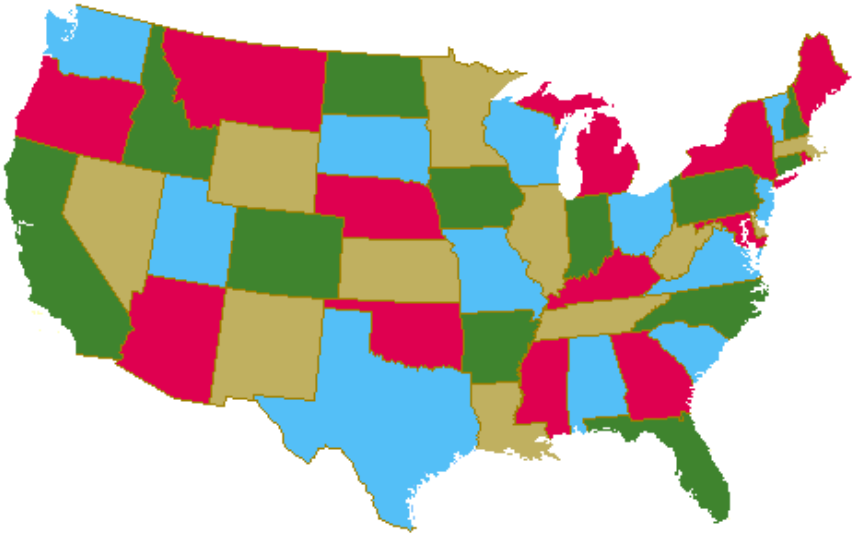


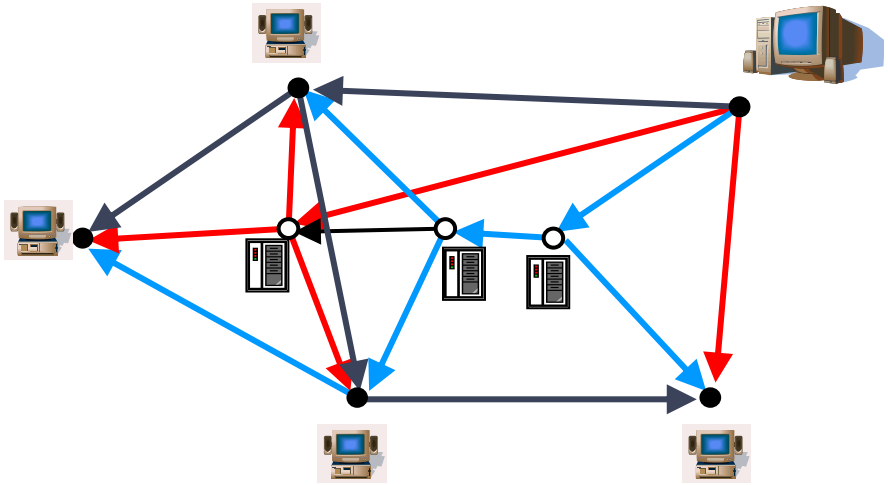


Computer networks, circuit design, data structures

# Topic 4: Graph Theory

How to color a map?





How to send data efficiently?

# The subjects covered in this course

- **Propositional logic** – language, essence and rules of reasoning using propositions.
- **Predicate logic** – reasoning with statements involving variables.
- **Sets** – the basis of every theory in computer science.
- **Mathematical reasoning** – recursive definitions and mathematical induction.
- **Relations** – one of the key concepts in many subjects on computer and computation. For example, a database is viewed as a set of relations and database query languages are constructed based on operations on relations and sets.
- **Graphs** – good example of discrete structures and one of the most useful models for computer scientists and engineers in solving problems.
- **Functions** – the special type of relations, one of the most important concepts in data structures, formal languages and automata, and analysis of algorithms.

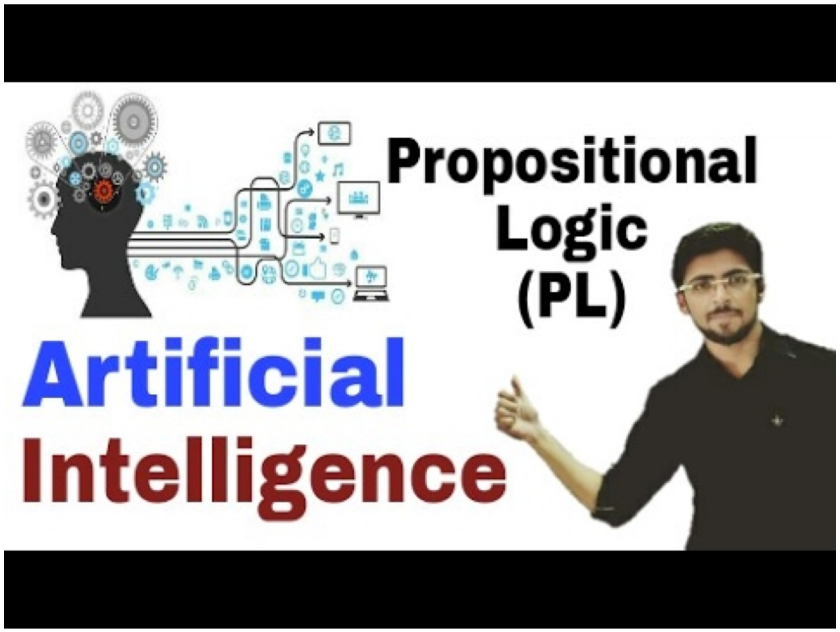
# Outline

- Propositional Logics
- Predicate Logics
- Rules of reference
- Proofs
- Applications

# Propositional Logics

# Introduction: Logic?

- Logic
  1) is the study of the logic relationships between objects
  2) forms the basis of all mathematical reasoning and all automated reasoning



# Introduction: PL?

- In Propositional Logic (a.k.a Propositional Calculus or Sentential Logic), the objects are called propositions
- **Definition**: A proposition is a statement that is either true or false, but not both
- We usually denote a proposition by a letter: *p*, *q*, *r*, *s*, …

# Propositions

- Propositional logic operates with statements. Statements could be true or false and are called **propositions**.
- *Is the sentence proposition?*

| | |
|---|---|
| Richmond is the capital of Virginia. | Yes (True) |
| $2 + 3 = 7$. | Yes (False) |
| Open the door. | No |
| $5 + 7 < 10$. | Yes (False) |
| The moon is a satellite of the earth. | Yes (True) |
| $x + 5 = 7$. | No |
| $x + 5 > 9$ for every real number $x$. | Yes (False) |

# Propositions: Examples

- The following are propositions
  - Today is Monday
  - The grass is wet
  - It is raining
- The following are not propositions
  - C++ is the best language *Opinion*
  - When is the pretest? *Interrogative*
  - Do your homework *Imperative*

# Are these propositions?

- 2+2=5
- Every integer is divisible by 12
- Microsoft is an excellent company

# Logical connectives

- Connectives are used to create a compound proposition from two or more propositions
  - Negation (denote $\sim$ or $\neg$ or !) $\neg$
  - And or logical conjunction (denoted $\wedge$) $\wedge$
  - Or or logical disjunction (denoted $\vee$) $\vee$
  - XOR or exclusive or (denoted $\oplus$) $\oplus$
  - Implication (denoted $\Rightarrow$ or $\rightarrow$)

    $\Rightarrow$, $\rightarrow$
  - Biconditional (denoted $\Leftrightarrow$ or $\leftrightarrow$)

    $\Leftrightarrow$, $\leftrightarrow$
- We define the meaning (semantics) of the logical connectives using truth tables

# Truth Tables

- Truth tables are used to show/define the relationships between the truth values of
  - the individual propositions and
  - the compound propositions based on them

| $p$ | $q$ | $p \wedge q$ | $p \vee q$ | $p \oplus q$ | $p \Rightarrow q$ | $p \Leftrightarrow q$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 |

# Logical Connective: Implication

- The implication of $p \rightarrow q$ can be also read as
  - If $p$ then $q$
  - $p$ implies $q$
  - If $p$, $q$
  - $p$ only if $q$
  - $q$ if $p$
  - $q$ when $p$
  - $q$ whenever $p$
  - $q$ follows from $p$
  - $p$ is a sufficient condition for $q$ ($p$ is sufficient for $q$)
  - $q$ is a necessary condition for $p$ ($q$ is necessary for $p$)

# Logical Connective: Implication

- Examples
  - If you buy you air ticket in advance, it is cheaper.
  - If $x$ is an integer, then $x^2 \geq 0$.
  - If it rains, the grass gets wet.
  - If the sprinklers operate, the grass gets wet.
  - If 2+2=5, then all unicorns are pink.

# Exercise: Which of the following implications is true?

- If -1 is a positive number, then 2+2=5

True. The premise is obviously false, thus no matter what the conclusion is, the implication holds.

- If -1 is a positive number, then 2+2=4

True. Same as above.

- If $\sin x = 0$, then $x = 0$

False. $x$ can be a multiple of $\pi$. If we let $x=2\pi$, then $\sin x=0$ but $x\neq 0$. The implication "if $\sin x = 0$, then $x = k\pi$, for some $k$" is true.

# Logical Connective: Biconditional

- The biconditional $p \leftrightarrow q$ can be equivalently read as
  - $p$ if and only if $q$
  - $p$ is a necessary and sufficient condition for $q$
  - if $p$ then $q$, and conversely
  - $p$ iff $q$ (Note typo in textbook, page 9, line 3)
- Examples
  - $x>0$ if and only if $x^2$ is positive
  - The alarm goes off iff a burglar breaks in
  - You may have pudding iff you eat your meat

# Exercise: Which of the following biconditionals is true?

- $x^2 + y^2 = 0$ if and only if $x=0$ and $y=0$

  True. Both implications hold

- $2 + 2 = 4$ if and only if $\sqrt{2}<2$

  True. Both implications hold.

- $x^2 \geq 0$ if and only if $x \geq 0$

  False. The implication "if $x \geq 0$ then $x^2 \geq 0$" holds.
  However, the implication "if $x^2 \geq 0$ then $x \geq 0$" is false.
  Consider x=-1.
  The hypothesis $(-1)^2=1 \geq 0$ but the conclusion fails.

# Converse, Inverse, Contrapositive

- For the proposition $P \longrightarrow Q$,
  - the proposition $\neg P \longrightarrow \neg Q$ is called its **inverse**,
  - the proposition is $Q \longrightarrow P$ called its **converse**,
  - the proposition $\neg Q \longrightarrow \neg P$ is called its **contrapositive**.

- The inverse and converse of a proposition are not necessarily logically equivalent to the proposition.

- The contrapositive of a proposition is always logically equivalent to the proposition.

# Converse, Inverse, Contrapositive

- **Example:** for the proposition "If it rains,  then I get wet",
  - Inverse: If does not rain, then I don’t get wet.
  - Converse: If I get wet,  then it rains.
  - Contrapositive: If I don't get wet,  then it does not rain.

- Therefore, "If it rains,  then I get wet." and "If I don't get wet,  then it does not rain." are logically equivalent. If one is true then the other is also true, and vice versa.

# Constructing Truth Tables

- Construct the truth table for the following compound proposition

$$(( p \wedge q ) \vee \neg q )$$

| $p$ | $q$ | $p \wedge q$ | $\neg q$ | $(( p \wedge q ) \vee \neg q )$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |

# Precedence of Logical Operators

- As in arithmetic, an ordering is imposed on the use of logical operators in compound propositions
- However, it is preferable to use parentheses to disambiguate operators and facilitate readability

$$\neg p \vee q \wedge \neg r \quad \equiv \quad (\neg p) \vee (q \wedge (\neg r))$$

- To avoid unnecessary parenthesis, the following precedences hold:
  1. Negation ($\neg$)
  2. Conjunction ($\wedge$)
  3. Disjunction ($\vee$)
  4. Implication ($\rightarrow$)
  5. Biconditional ($\leftrightarrow$)

# Usefulness of Logic

- Logic is more precise than natural language
  - You may have cake or ice cream.
  - Can I have both?
  - If you buy your air ticket in advance, it is cheaper.
  - Are there or not cheap last-minute tickets?

- For this reason, logic is used for hardware and software specification
  - Given a set of logic statements, one can decide whether or not they are satisfiable (i.e., consistent), although this is a costly process…

# Bitwise Operations

- Computers represent information as bits (binary digits)
- A bit string is a sequence of bits
- The length of the string is the number of bits in the string
- Logical connectives can be applied to bit strings of equal length
- Example

| | |
|---|---|
| | 0110 1010 1101 |
| | 0101 0010 1111 |
| | _____ |
| Bitwise OR | 0111 1010 1111 |
| Bitwise AND | ... |
| Bitwise XOR | … |

# Logic in Programming: Example 1

- Say you need to define a conditional statement as follows:
  - Increment x if all of the following conditions hold: x > 0, x < 10, x=10
- You may try: `If (0<x<10 OR x==10) x++;`
- But this is not valid in C++ or Java. How can you modify this statement by using logical equivalence
- Answer: `If (x>0 AND x<=10) x++;`

# Logic in Programming: Example 2

- Say we have the following loop

```
While
   ((i<size AND A[i]>10) OR
    (i<size AND A[i]<0) OR
    (i<size AND (NOT (A[i]!=0 AND NOT (A[i]>=10)))))
```

- Is this a good code? Keep in mind:
  - Readability
  - Extraneous code is inefficient and poor style
  - Complicated code is more prone to errors and difficult to debug
  - Solution? Comes later…

# Propositional Equivalences: Introduction

- To manipulate a set of statements (here, logical propositions) for the sake of mathematical argumentation, an important step is to replace one statement with another equivalent statement (i.e., with the same truth value)
- Below, we discuss:
  - Terminology
  - Establishing logical equivalences using truth tables
  - Establishing logical equivalences using known laws (of logical equivalences)

# Terminology

- Definitions
  - A compound proposition that is always true, no matter what the truth values of the propositions that occur in it is called a tautology
  - A compound proposition that is always false is called a contradiction
  - A proposition that is neither a tautology nor a contradiction is a contingency
- Examples
  - A simple tautology is $p \vee \neg p$
  - A simple contradiction is $p \wedge \neg p$

# Logical Equivalences: Definition

- **Definition**: Propositions $p$ and $q$ are logically equivalent if $p \leftrightarrow q$ is a tautology.
- Informally, p and q are equivalent if whenever p is true, q is true, and vice versa
- Notation: $p \equiv q$ ($p$ is equivalent to $q$), $p \leftrightarrow q$, and $p \Leftrightarrow q$
- Alert: $\equiv$ is not a logical connective $\equiv$

# Logical Equivalences: Example 1

- Are the propositions $(p \rightarrow q)$ and $(\neg p \vee q)$ logically equivalent?
- To find out, we construct the truth tables for each:

| $p$ | $q$ | $p \rightarrow q$ | $\neg p$ | $\neg p \vee q$ |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 |

The two columns in the truth table are identical, thus we conclude that

$$(p \rightarrow q) \equiv (\neg p \vee q)$$

# Logical Equivalences: Example 2

- Show that $(p \rightarrow r) \vee (q \rightarrow r) \equiv (p \wedge q) \rightarrow r$

| $p$ | $q$ | $r$ | $p \rightarrow r$ | $q \rightarrow r$ | $(p \rightarrow r) \vee (q \rightarrow r)$ | $p \wedge q$ | $(p \wedge q) \rightarrow r$ |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

# Logical Equivalences: Cheat Sheet (1)

| Identities (Equivalences) | Name |
|---|---|
| $p \wedge T \equiv p$<br>$p \vee F \equiv p$ | Identity laws |
| $p \vee T \equiv T$<br>$p \wedge F \equiv F$ | Domination laws |
| $p \vee p \equiv p$<br>$p \wedge p \equiv p$ | Idempotent laws |
| $\neg(\neg p) \equiv p$ | Double negation law |
| $p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$ | Commutative laws |
| $(p \vee q) \vee r \equiv p \vee (q \vee r)$<br>$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$ | Associative laws |
| $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$<br>$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$ | Distributive laws |
| $\neg(p \wedge q) \equiv \neg p \vee \neg q$<br>$\neg(p \vee q) \equiv \neg p \wedge \neg q$ | De Morgan's laws |
| $p \vee (p \wedge q) \equiv p$<br>$p \wedge (p \vee q) \equiv p$ | Absorption laws |
| $p \vee \neg p \equiv T$<br>$p \wedge \neg p \equiv F$ | Negation laws |

# Logical Equivalences: Cheat Sheet (2)

**Logical Equivalences Involving Conditional Statements.**

$p \rightarrow q \equiv \neg p \vee q$
$p \rightarrow q \equiv \neg q \rightarrow \neg p$
$p \vee q \equiv \neg p \rightarrow q$
$p \wedge q \equiv \neg(p \rightarrow \neg q)$
$\neg(p \rightarrow q) \equiv p \wedge \neg q$
$(p \rightarrow q) \wedge (p \rightarrow r) \equiv p \rightarrow (q \wedge r)$
$(p \rightarrow r) \wedge (q \rightarrow r) \equiv (p \vee q) \rightarrow r$
$(p \rightarrow q) \vee (p \rightarrow r) \equiv p \rightarrow (q \vee r)$
$(p \rightarrow r) \vee (q \rightarrow r) \equiv (p \wedge q) \rightarrow r$

# Using Logical Equivalences: Example 1

- Logical equivalences can be used to construct additional logical equivalences
- Example: Show that $(p \wedge q) \rightarrow q$ is a tautology

| | | |
|---|---|---|
| 0. | $(p \wedge q) \rightarrow q$ | |
| 1. | $\equiv \neg(p \wedge q) \vee q$ | Implication Law on 0 |
| 2. | $\equiv (\neg p \vee \neg q) \vee q$ | De Morgan's Law (1st) on 1 |
| 3. | $\equiv \neg p \vee (\neg q \vee q)$ | Associative Law on 2 |
| 4. | $\equiv \neg p \vee 1$ | Negation Law on 3 |
| 5. | $\equiv 1$ | Domination Law on 4 |

# Using Logical Equivalences: Example 3

- Example (Exercise 17)*: Show that $\neg(p \leftrightarrow q) \equiv (p \leftrightarrow \neg q)$
- Sometimes it helps to start with the second proposition ($p \leftrightarrow \neg q$)

| | | |
|---|---|---|
| 0. | $(p \leftrightarrow \neg q)$ | |
| 1. | $\equiv (p \rightarrow \neg q) \wedge (\neg q \rightarrow p)$ | Equivalence Law on 0 |
| 2. | $\equiv (\neg p \vee \neg q) \wedge (q \vee p)$ | Implication Law on 1 |
| 3. | $\equiv \neg(\neg((\neg p \vee \neg q) \wedge (q \vee p)))$ | Double negation on 2 |
| 4. | $\equiv \neg(\neg(\neg p \vee \neg q) \vee \neg(q \vee p))$ | De Morgan's Law… |
| 5. | $\equiv \neg((p \wedge q) \vee (\neg q \wedge \neg p))$ | De Morgan's Law |
| 6. | $\equiv \neg((p \vee \neg q) \wedge (p \vee \neg p) \wedge (q \vee \neg q) \wedge (q \vee \neg p))$ | Distribution Law |
| 7. | $\equiv \neg((p \vee \neg q) \wedge (q \vee \neg p))$ | Identity Law |
| 8. | $\equiv \neg((q \rightarrow p) \wedge (p \rightarrow q))$ | Implication Law |
| 9. | $\equiv \neg(p \leftrightarrow q)$ | Equivalence Law |

*See Table 8 (p 25) but you are not allowed to use the table for the proof

# Using Logical Equivalences: Example 3

- Show that $\neg(q \rightarrow p) \vee (p \wedge q) \equiv$ q

| | | |
|---|---|---|
| 0. | $\neg(q \rightarrow p) \vee (p \wedge q)$ | |
| 1. | $\equiv \neg(\neg q \vee p) \vee (p \wedge q)$ | Implication Law |
| 2. | $\equiv (q \wedge \neg p) \vee (p \wedge q)$ | De Morgan's & Double negation |
| 3. | $\equiv (q \wedge \neg p) \vee (q \wedge p)$ | Commutative Law |
| 4. | $\equiv q \wedge (\neg p \vee p)$ | Distributive Law |
| 5. | $\equiv q \wedge 1$ | Identity Law |
| | $\equiv q$ | Identity Law |

# Logic in Programming: Example 2 (revisited)

- Recall the loop

```
While
    ((i<size AND A[i]>10) OR
     (i<size AND A[i]<0) OR
     (i<size AND (NOT (A[i]!=0 AND NOT (A[i]>=10)))))
```

- Now, using logical equivalences, simplify it!
- Using De Morgan's Law and Distributivity

```
While ((i<size) AND
       ((A[i]>10 OR A[i]<0) OR
          (A[i]==0 OR A[i]>=10)))
```

- Noticing the ranges of the 4 conditions of A[i]

```
While ((i<size) AND (A[i]>=10 OR A[i]<=0))
```

# Programming Pitfall Note

- In C, C++ and Java, applying the commutative law is not such a good idea.
- For example, consider accessing an integer array A of size n:

```
if (i<n && A[i]==0) i++;
```

is not equivalent to

```
if (A[i]==0 && i<n) i++;
```

# Predicate Logic

# Introduction

- Consider the statements:

$$x>3,\ x=y+3,\ x+y=z$$

- The symbols >, +, = denote relations between $x$ and 3, $x$, $y$, and 4, and $x$,$y$, and $z$, respectively
- These relations may hold or not hold depending on the values that $x$, $y$, and $z$ may take.
- A **predicate** is a property that is affirmed or denied about the subject (in logic, we say '**variable**' or '**argument**') of a statement
- Consider the statement : '$x$ is greater than 3'
  - '$x$' is the subject
  - 'is greater than 3' is the predicate

# Propositional Functions (1)

- To write in Predicate Logic '*x* is greater than 3'
  - We introduce a functional symbol for the **predicate** and
  - Put the subject as an **argument** (to the functional symbol): $P(x)$
- Terminology
  - $P(x)$ is a statement
  - $P$ is a predicate or propositional function
  - $x$ as an argument

# Propositional Functions (2)

- Examples:
  - Father(x): unary predicate
  - Brother(x.y): binary predicate
  - Sum(x,y,z): ternary predicate
  - P(x,y,z,t): n-ary predicate

# Propositional Functions (3)

- **Definition:** A statement of the form $P(x_1,x_2,\ldots, x_n)$ is the value of the propositional symbol $P$.
- Here: $(x_1,x_2,\ldots, x_n)$ is an $n$-tuple and $P$ is a predicate
- We can think of a propositional function as a function that
  - Evaluates to true or false
  - Takes one or more arguments
  - Expresses a predicate involving the argument(s)
  - Becomes a proposition when values are assigned to the arguments

# Propositional Functions: Example

- Let $Q(x,y,z)$ denote the statement '$x^2+y^2=z^2$'
  - What is the truth value of $Q(3,4,5)$?

    $Q(3,4,5)$ is true

  - What is the truth value of $Q(2,2,3)$?

    $Q(2,3,3)$ is false

  - How many values of $(x,y,z)$ make the predicate true?

    There are infinitely many values that make the proposition true, how many right triangles are there?

# Universe of Discourse

- Consider the statement '$x>3$', does it make sense to assign to $x$ the value 'blue'?
- Intuitively, the **universe of discourse** is the set of all things we wish to talk about; that is the set of all objects that we can sensibly assign to a variable in a propositional function.
- What would be the universe of discourse for the propositional function below be:

Enrolled ICSI 521(x)='x is enrolled in ICSI 521'

# Universe of Discourse: Multivariate functions

- Each variable in an *n*-tuple (i.e., each argument) may have a different universe of discourse
- Consider an *n*-ary predicate *P*:

  *P*(*r*,*g*,*b*,*c*)= 'The *rgb*-values of the color *c* is (*r*,*g*,*b*)'
- Example, what is the truth value of
  - *P*(*255*,*0*,*0*,*red*)
  - *P*(*0*,*0*,*255*,*green*)
- What are the universes of discourse of (*r*,*g*,*b*,*c*)?

# Quantifiers: Introduction

- The statement '*x*>*3*' is not a proposition
- It becomes a proposition
  - When we assign values to the argument: '4>3' is false, '2<3' is true, or
  - When we quantify the statement
- Two quantifiers
  - Universal quantifier ∀
    $\forall$

    the proposition is true for all possible values in the universe of discourse
  - Existential quantifier ∃
    $\exists$

    the proposition is true for some value(s) in the universe of discourse

# Universal Quantifier: Definition

- **Definition**: The universal quantification of a predicate $P(x)$ is the proposition '$P(x)$ is true for all values of $x$ in the universe of discourse.' We use the notation: $\forall\, x\, P(x)$, which is read 'for all $x$'.
- If the universe of discourse is finite, say $\{n_1, n_2, \ldots, n_k\}$, then the universal quantifier is simply the conjunction of the propositions over all the elements

$$\forall\, x\, P(x) \Leftrightarrow P(n_1) \wedge P(n_2) \wedge \ldots \wedge P(n_k)$$

# Universal Quantifier: Example 1

- Let $P(x)$: '$x$ must take a discrete mathematics course' and $Q(x)$: '$x$ is a CS student.'
- The universe of discourse for both $P(x)$ and $Q(x)$ is all UNL students.
- Express the statements:
  - "Every CS student must take a discrete mathematics course."

    $\forall x\, Q(x) \rightarrow P(x)$
  - "Everybody must take a discrete mathematics course or be a CS student." $\forall x\ (P(x) \vee Q(x))$
  - "Everybody must take a discrete mathematics course and be a CS student." $\forall x\, (P(x) \wedge Q(x))$

  Are these statements true or false?

# Universal Quantifier: Example 2

- Express the statement: ‘for every $x$ and every $y$, $x+y>10$’
- Answer:
  - Let $P(x,y)$ be the statement $x+y>10$
  - Where the universe of discourse for $x$, $y$ is the set of integers
  - The statement is: $\forall x \, \forall y \, P(x,y)$
- Shorthand: $\forall x,y \, P(x,y)$

# Existential Quantifier: Definition

- **Definition**: The existential quantification of a predicate $P(x)$ is the proposition 'There exists a value x in the universe of discourse such that $P(x)$ is true.' We use the notation: $\exists x\, P(x)$, which is read 'there exists $x$'.
- If the universe of discourse is finite, say $\{n_1, n_2, \ldots, n_k\}$, then the existential quantifier is simply the disjunction of the propositions over all the elements

$$\exists x\, P(x) \Leftrightarrow P(n_1) \vee P(n_2) \vee \ldots \vee P(n_k)$$

# Existential Quantifier: Example 1

- Let $P(x,y)$ denote the statement 'x+y=5'
- What does the expression $\exists x \, \exists y \, P(x,y)$ mean?
- Which universe(s) of discourse make it true?

# Existential Quantifier: Example 2

- Express the statement: 'there exists a real solution to $ax^2+bx-c=0$'
- Answer:
  - Let $P(x)$ be the statement $x= \frac{-b \pm \sqrt{b^2-4ac}}{2a}$
  - Where the universe of discourse for $x$ is the set of real numbers. Note here that $a$, $b$, $c$ are fixed constants.
  - The statement can be expressed as $\exists x\, P(x)$
- What is the truth value of $\exists x\, P(x)$?
  - It is false. When $b^2<4ac$, there are no real number x that can satisfy the predicate
- What can we do so that $\exists x\, P(x)$is true?
  - Change the universe of discourse to the complex numbers $\mathbb{C}$

# Quantifiers: Truth values

- In general, when are quantified statements true or false?

| Statement | True when… | False when… |
|---|---|---|
| $\forall x\, P(x)$ | $P(x)$ is true for every $x$ | There is an $x$ for which $P(x)$ is false |
| $\exists x\, P(x)$ | There is an $x$ for which $P(x)$ is true | $P(x)$ is false for every $x$ |

# Mixing quantifiers (1)

- Existential and universal quantifiers can be used together to quantify a propositional predicate. For example:

$$\forall x \, \exists y \, P(x,y)$$

  is perfectly valid
- Alert:
  - The quantifiers must be read from left to right
  - The order of the quantifiers is important
  - $\forall x \, \exists y \, P(x,y)$ is not equivalent to $\exists y \, \forall x \, P(x,y)$

# Mixing quantifiers (2)

- Consider
  - $\forall x \exists y$ *Loves* $(x,y)$: Everybody loves somebody
  - $\exists y \forall x$ *Loves*$(x,y)$: There is someone loved by everyone
- The two expressions do not mean the same thing
- $(\exists y \forall x$ *Loves*$(x,y)) \rightarrow (\forall x \exists y$ *Loves* $(x,y))$ but the converse does not hold
- However, you can commute similar quantifiers
  - $\forall x \forall y P(x,y)$ is equivalent to $\forall y \forall x P(x,y)$ (thus, $\forall x,y$ $P(x,y)$)
  - $\exists x \exists y P(x,y)$ is equivalent to $\exists y \exists x P(x,y)$ (thus $\exists$x,$y$ $P(x,y)$)

# Mixing Quantifiers: Truth values

| Statement | True when... | False when... |
|---|---|---|
| $\forall x \forall y$ $P(x,y)$ | $P(x,y)$ is true for every pair $x,y$ | There is at least one *pair* $x,y$ for which $P(x,y)$ is false |
| $\forall x \exists y$ $P(x,y)$ | For every x, there is a $y$ for which $P(x,y)$ is true | There is an $x$ for which $P(x,y)$ is false for every $y$ |
| $\exists x \forall y$ $P(x,y)$ | There is an $x$ for which $P(x,y)$ is true for every $y$ | For every $x$, there is a $y$ for which $P(x,y)$ is false |
| $\exists x \exists y P(x,y)$ | There is at least one pair $x,y$ for which $P(x,y)$ is true | $P(x,y)$ is false for every pair $x,y$ |

# Universal Quantifier: Definition

- **Definition**: The universal quantification of a predicate $P(x)$ is the proposition '$P(x)$ is true for all values of $x$ in the universe of discourse.' We use the notation: $\forall\, x\, P(x)$, which is read 'for all $x$'.
- If the universe of discourse is finite, say $\{n_1, n_2, \ldots, n_k\}$, then the universal quantifier is simply the conjunction of the propositions over all the elements

$$\forall\, x\, P(x) \Leftrightarrow P(n_1) \wedge P(n_2) \wedge \ldots \wedge P(n_k)$$

# Existential Quantifier: Definition

- **Definition**: The existential quantification of a predicate $P(x)$ is the proposition 'There exists a value x in the universe of discourse such that $P(x)$ is true.' We use the notation: $\exists x\, P(x)$, which is read 'there exists $x$'.
- If the universe of discourse is finite, say $\{n_1, n_2, \ldots, n_k\}$, then the existential quantifier is simply the disjunction of the propositions over all the elements

$$\exists x\, P(x) \Leftrightarrow P(n_1) \vee P(n_2) \vee \ldots \vee P(n_k)$$

# Quantifiers: Truth values

- In general, when are quantified statements true or false?

| Statement | True when… | False when… |
|---|---|---|
| $\forall x\, P(x)$ | $P(x)$ is true for every $x$ | There is an $x$ for which $P(x)$ is false |
| $\exists x\, P(x)$ | There is an $x$ for which $P(x)$ is true | $P(x)$ is false for every $x$ |

# Mixing quantifiers (1)

- Existential and universal quantifiers can be used together to quantify a propositional predicate. For example:

$$\forall x \, \exists y \, P(x,y)$$

  is perfectly valid
- Alert:
  - The quantifiers must be read from left to right
  - The order of the quantifiers is important
  - $\forall x \, \exists y \, P(x,y)$ is not equivalent to $\exists y \, \forall x \, P(x,y)$

# Mixing quantifiers (2)

- Consider
  - $\forall x \, \exists y \, Loves \, (x,y)$: Everybody loves somebody
  - $\exists y \, \forall x \, Loves(x,y)$: There is someone loved by everyone
- The two expressions do not mean the same thing
- $(\exists y \, \forall x \, Loves(x,y)) \rightarrow (\forall x \, \exists y \, Loves \, (x,y))$ but the converse does not hold
- However, you can commute similar quantifiers
  - $\forall x \, \forall y \, P(x,y)$ is equivalent to $\forall y \, \forall x \, P(x,y)$ (thus, $\forall x,y \, P(x,y)$)
  - $\exists x \, \exists y \, P(x,y)$ is equivalent to $\exists y \, \exists x \, P(x,y)$ (thus $\exists x,y \, P(x,y)$)

# Mixing Quantifiers: Truth values

| Statement | True when... | False when... |
|---|---|---|
| $\forall x \forall y$ $P(x,y)$ | $P(x,y)$ is true for every pair $x,y$ | There is at least one *pair* $x,y$ for which $P(x,y)$ is false |
| $\forall x \exists y$ $P(x,y)$ | For every x, there is a $y$ for which $P(x,y)$ is true | There is an $x$ for which $P(x,y)$ is false for every $y$ |
| $\exists x \forall y$ $P(x,y)$ | There is an $x$ for which $P(x,y)$ is true for every $y$ | For every $x$, there is a $y$ for which $P(x,y)$ is false |
| $\exists x \exists y P(x,y)$ | There is at least one pair $x,y$ for which $P(x,y)$ is true | $P(x,y)$ is false for every pair $x,y$ |

# Mixing Quantifiers: Example (1)

- Express, in predicate logic, the statement that there is an infinite number of integers
- Answer:

  – Let $P(x,y)$ be the statement that $x<y$

  – Let the universe of discourse be the integers, $Z$

  – The statement can be expressed by the following

$$\forall x \, \exists y \, P(x,y)$$

# Mixing Quantifiers: Example (2)

- Express the *commutative law of addition* for $R$
- We want to express that for every pair of reals, $x,y,$ the following holds: $x+y=y+x$
- Answer:

  – Let $P(x,y)$ be the statement that $x+y$

  – Let the universe of discourse be the reals, $R$

  – The statement can be expressed by the following

$$\forall x \, \forall y \, (P(x,y) \Leftrightarrow P(y,x))$$

Alternatively, $\forall x \, \forall y \, (x+y = y+x)$

# Mixing Quantifiers: Example (3)

- Express the multiplicative *law* for nonzero reals $R \setminus \{0\}$
- We want to express that for every real number $x$, there exists a real number $y$ such that $xy$=1
- Answer:

$$\forall x \, \exists y \, (xy = 1)$$

# Mixing Quantifiers: Example (4)
## false mathematical statement

- Does commutativity for substraction hold over the reals?
- That is: does $x$-$y$=$y$-$x$ for all pairs $x$,$y$ in $R$?
- Express using quantifiers

$$\forall x \, \forall y \, (x\text{-}y = y\text{-}x)$$

# Mixing Quantifiers: Example (5)

- Express the statement as a logical expression: “There is a number x such that when it is added to any number, the result is that number and if it is multiplied by any number, the result is x” as a logical expression
- Answer:
  - Let $P(x,y)$ be the expression “$x+y=y$”
  - Let $Q(x,y)$ be the expression “$xy=x$”
  - The universe of discourse is $\mathbb{N},\mathbb{Z},\mathbb{R},\mathbb{Q}$ *(but not* $\mathbb{Z}^+$*)*
  - Then the expression is:

$$\exists x \, \forall y \, P(x,y) \wedge Q(x,y)$$

Alternatively: $\exists x \, \forall y \, (x+y=y) \wedge (xy = x)$

# Binding Variables

- When a quantifier is used on a variable $x$, we say that $x$ is bound
- If no quantifier is used on a variable in a predicate statement, the variable is called free
- Examples
  - In $\exists x \forall y P(x,y)$, both $x$ and $y$ are bound
  - In $\forall x P(x,y)$, $x$ is bound but $y$ is free
- A statement is called a well-formed formula, when all variables are properly quantified

# Binding Variables: Scope

- The set of all variables bound by a common quantifier is called the scope of the quantifier
- For example, in the expression $\exists x,y \forall z P(x,y,z,c)$
  - What is the scope of existential quantifier?
  - What is the scope of universal quantifier?
  - What are the bound variables?
  - What are the free variables?
  - Is the expression a well-formed formula?

# Negation

- We can use negation with quantified expressions as we used them with propositions
- **Lemma**: Let $P(x)$ be a predicate. Then the followings hold:

$$\neg(\forall x\, P(x)) \equiv \exists x\, \neg P(x)$$

$$\neg(\exists x\, P(x)) \equiv \forall x\, \neg P(x)$$

- This is essentially the quantified version of De Morgan's Law (when the universe of discourse is finite, this is exactly De Morgan's Law)

# Negation: Truth

Truth Values of Negated Quantifiers

| Statement | True when… | False when… |
|---|---|---|
| $\neg\exists x\, P(x) \equiv \forall x\, \neg P(x)$ | $P(x)$ is false for every $x$ | There is an $x$ for which $P(x)$ is true |
| $\neg\forall x\, P(x) \equiv \exists x\, \neg P(x)$ | There is an $x$ for which $P(x)$ is false | $P(x)$ is true for every $x$ |

# Prolog (1)

- Prolog (Programming in Logic) is a programming language based on (a restricted form of) Predicate Logic (a.k.a. Predicate Calculus and FOL)
- It was developed by the logicians of the Artificial Intelligence community for symbolic reasoning

# Prolog (2)

- Prolog allows the users to express facts and rules
- Facts are propositional functions: student(mia), enrolled(mia,cse235), instructor(patel,cse235), etc.
- Rules are implications with conjunctions:

  teaches(X,Y) :- instructor(X,Z), enrolled(Y,Z)
- Prolog answers queries such as:

  ?enrolled(mia,cse235)

  ?enrolled(X,cse476)

  ?teaches(X,mia)

  by binding variables and doing theorem proving (i.e., applying inference rules) as we will see in Section 1.5

# English into Logic

- Logic is more precise than English
- Transcribing English into Logic and vice versa can be tricky
- When writing statements with quantifiers, usually the correct meaning is conveyed with the following combinations:

**Use $\forall$ with $\Rightarrow$**

$\forall x\ Lion(x) \Rightarrow Fierce(x)$: Every lion is fierce

$\forall x\ Lion(x) \wedge Fierce(x)$: Everyone is a lion and everyone is fierce

**Use $\exists$ with $\wedge$**

$\exists\ x\ Lion(x) \wedge Vegan(x)$: Holds when you have at least one vegan lion

$\exists\ x\ Lion(x) \Rightarrow Vegan(x)$: Holds when you have vegan people in the universe of discourse (even though there is no vegan lion in the universe of discourse )

# More exercises (1)

- Rewrite the following expression, pushing negation inward:

$$\neg\, \forall x\, (\exists\, y\, \forall z\, P(x,y,z) \wedge \exists\, z\, \forall y\, P(x,y,z))$$

- Answer:

$$\exists x\, (\forall y\, \exists z\, \neg P(x,y,z) \vee \forall z\, \exists y\, \neg P(x,y,z))$$

# More Exercises (2)

- Let P(x,y) denote ‘x is a factor of y’ where
  - $x \in \{1,2,3,\ldots\}$ and $y \in \{2,3,4,\ldots\}$
- Let $Q(x,y)$ denote:
  - $\forall x,y \ [P(x,y) \rightarrow (x=y) \vee (x=1)]$
- Question: When is $Q(x,y)$ true?

# Alert…

- Some students wonder if:

$$\forall x,y\, P(x,y) \equiv (\forall x\, P(x,y)) \wedge (\forall y\, P(x,y))$$

- This is certainly not true.
  - In the left-hand side, both x,y are bound.
  - In the right-hand side,
    - In the first predicate, *x* is bound and *y* is free
    - In the second predicate, *y* is bound and *x* is free
    - Thus, the left-hand side is a proposition, but the right-hand side is not. They cannot be equivalent
- All variables that occur in a propositional function must be bound to turn it into a proposition

# Rules of Inference

# Motivation (1)

- "Mathematical proofs, like diamonds, are hard and clear, and will be touched with nothing but strict reasoning." *-John Locke*
- Mathematical proofs are, in a sense, the only true knowledge we have
- They provide us with a guarantee as well as an explanation (and hopefully some insight)

# Motivation (2)

- Mathematical proofs are necessary in CS
  - You must always (try to) prove that your algorithm
    - terminates
    - is sound, complete, optimal
    - finds optimal solution
  - You may also want to show that it is more efficient than another method
  - Proving certain properties of data structures may lead to new, more efficient or simpler algorithms
  - Arguments may entail assumptions. You may want to prove that the assumptions are valid

# Concepts

- A **theorem** is a statement that can be shown to be true (via a proof)
- A **proof** is a sequence of statements that form an argument
- A **corollary** is a theorem that can be established from theorem that has just been proven
- A **conjecture** is a statement whose truth value is unknown
- The **rules of inference** are the means used to draw conclusions from other assertions, and to derive an argument or a proof

# Theorems: Example

- Theorem
  - Let $a$, $b$, and $c$ be integers. Then
    - If $a|b$ and $a|c$ then $a|(b+c)$
    - If $a|b$ then $a|bc$ for all integers $c$
    - If $a|b$ and $b|c$, then $a|c$
- Corollary:
  - If $a$, $b$, and $c$ are integers such that $a|b$ and $a|c$, then $a|mb+nc$ whenever $m$ and $n$ are integers
- What is the assumption? What is the conclusion?

# Rules of Inference: Modus Ponens

- Intuitively, modus ponens (or law of detachment) can be described as the inference:

  p implies q; p is true; therefore q holds

- In logic terminology, modus ponens is the tautology:

$$(p \wedge (p \rightarrow q)) \rightarrow q$$

- Note: 'therefore' is sometimes denoted $\therefore$, so we have:

$$p \rightarrow q \equiv p \therefore q$$

# Rules of Inference: Addition

- Addition involves the tautology

$$p \rightarrow (p \vee q)$$

- Intuitively,
  - if we know that p is true
  - we can conclude that either p or q are true (or both)
- In other words: $p \therefore (p \vee q)$
- Example: I read the newspaper today, therefore I read the newspaper or I ate custard
  - Note that these are <u>not</u> mutually exclusive

# Rules of Inference: Simplification

- Simplification is based on the tautology

$$(p \wedge q) \rightarrow p$$

- So we have: $(p \wedge q) \therefore p$
- Example: Prove that if $0 < x < 10$, then $x \geq 0$

1. $0 < x < 10 \equiv (0 < x) \wedge (x < 10)$
2. $(x > 0) \wedge (x < 10) \rightarrow (x > 0)$ by simplification
3. $(x > 0) \rightarrow (x > 0) \vee (x = 0)$ by addition
4. $(x > 0) \vee (x = 0) \equiv (x \geq 0)$ Q.E.D.

QED= Latin word for “quod erat demonstrandum” meaning “that which was to be demonstrated” or “that which was to be shown”.

# Rules of inference: Conjunction

- The conjunction is almost trivially intuitive. It is based on the following tautology:

$$((p) \wedge (q)) \rightarrow (p \wedge q)$$

- Note the subtle difference though:
  - On the left-hand side, we independently know $p$ and $q$ to be true
  - Therefore, we conclude, on the right-hand side, that a logical conjunction is true

# Rules of Inference: Modus Tollens

- Similar to the modus ponens, modus tollens is based on the following tautology

$$(\neg q \wedge (p \rightarrow q)) \rightarrow \neg p$$

- In other words:
  - If we know that $q$ is not true
  - And that p implies $q$
  - Then we can conclude that $p$ does not hold either
- Example
  - If you are New York State citizen student, then you pay in-state tuition fee at SUNY.
  - Tom pay out-state tuition fee at SUNY.
  - Therefore we can conclude that Tom is not a New York State citizen student.

# Rules of Inference: Contrapositive

- The contrapositive is the following tautology

$$(p \rightarrow q) \rightarrow (\neg q \rightarrow \neg p)$$

- Usefulness
  - If you are having trouble proving the p implies q in a direct manner
  - You can try to prove the contrapositive instead!

# Rules of Inference: Hypothetical Syllogism

- Hypothetical syllogism is based on the following tautology

$$((p \rightarrow q) \wedge (q \rightarrow r)) \rightarrow (p \rightarrow r)$$

- Essentially, this shows that the rules of inference are, in a sense, transitive
- Example:
  - If you don't get a job, you won't have money
  - If you don't have money, you will starve.
  - Therefore, if you don't get a job, you'll starve

# Rules of Inference: Disjunctive Syllogism

- A disjunctive syllogism is formed on the basis of the tautology

$$((p \vee q) \wedge \neg p) \rightarrow q$$

- Reading this in English, we see that
  - If either *p* or *q* hold and we know that p does not hold
  - Then we can conclude that *q* must hold
- Example
  - The sky is either blue or grey
  - Well it isn't blue
  - Therefore, the sky is grey

# Rules of Inference: Resolution

- For resolution, we have the following tautology

$$((p \vee q) \wedge (\neg p \vee r)) \rightarrow (q \vee r)$$

- Essentially,
  - If we have two true disjunctions that have mutually exclusive propositions
  - Then we can conclude that the disjunction of the two non-mutually exclusive propositions is true

# Proofs: Example (1)

**Theorem**: *The sum of two odd integers is even*

- Let n, m be two odd integers.
- Every odd integer x can be written as $x=2k+1$ for some integer k
- Therefore, let $n=2k_1+1$ and $m=2k_2+1$
- Consider

$n+m = (2k_1+1)+(2k_2+1)$

$= 2k_1+ 2k_2+1+1$

*Associativity/Commutativity*

$= 2k_1+ 2k_2+2$

*Algebra*

$= 2(k_1+ k_2+1)$

*Factoring*

- By definition $2(k_1+k_2+1)$ is even, therefore n+m is even *QED*

# Proofs: Example (2)

- Assume that the statements below hold:
  - $(p \rightarrow q)$
  - $(r \rightarrow s)$
  - $(r \vee p)$
- Assume that q is false
- Show that s must be true

# Proofs: Example (2)

1. $(p \rightarrow q)$
2. $(r \rightarrow s)$
3. $(r \vee p)$
4. $\neg q$
5. $(\neg q \wedge (p \rightarrow q)) \rightarrow \neg p$ by modus tollens on 1 + 4
6. $(r \vee p) \wedge \neg p) \rightarrow r$ by disjunctive syllogism 3 + 6
7. $(r \wedge (r \rightarrow s)) \rightarrow s$ by modus ponens 2 + 6

# If and Only If

- If you are asked to show an equivalence

$$p \leftrightarrow q \text{ “if an only if”}$$

- You must show an implication in both directions
- That is, you can show (independently or via the same technique) that $(p \rightarrow q)$ and $(q \rightarrow p)$
- Example
  - Show that x is odd iff $x^2+2x+1$ is even

# Example (iff)

| | | |
|---|---|---|
| x is odd | $\leftrightarrow x=2k+1, k \in Z$ | *by definition* |
| | $\leftrightarrow x+1 = 2k+2$ | *algebra* |
| | $\leftrightarrow x+1 = 2(k+1)$ | *factoring* |
| | $\leftrightarrow x+1$ is even | *by definition* |
| | $\leftrightarrow (x+1)^2$ is even | *Since x is even iff $x^2$ is even* |
| | $\leftrightarrow x^2+2x+1$ is even | *algebra* |
| | | QED |

# Fallacies (1)

- Even a bad example is worth something: it teaches us what not to do
- There are three common mistakes (at least..).
- These are known as fallacies
  1. Fallacy of affirming the conclusion

$$(q \wedge (p \rightarrow q)) \rightarrow p$$

  2. Fallacy of denying the hypothesis

$$(\neg p \wedge (p \rightarrow q)) \rightarrow \neg q$$

  3. Circular reasoning. Here you use the conclusion as an assumption, avoiding an actual proof

# Little Reminder

- Affirming the antecedent: Modus ponens

$$(p \wedge (p \rightarrow q)) \rightarrow q$$

- Denying the consequent: Modus Tollens

$$(\neg q \wedge (p \rightarrow q)) \rightarrow \neg p$$

- Affirming the conclusion: Fallacy

$$(q \wedge (p \rightarrow q)) \rightarrow p$$

- Denying the hypothesis: Fallacy

$$(\neg p \wedge (p \rightarrow q)) \rightarrow \neg q$$

# Fallacies (2)

- Sometimes, bad proofs arise from illegal operations rather than poor logic.
- Consider the bad proof 2=1
- Let: $a = b$

| | |
|---|---|
| $a^2 = ab$ | *Multiply both sides by a* |
| $a^2 + a^2 - 2ab = ab + a^2 - 2ab$ | *Add $a^2 - 2ab$ to both sides* |
| $2(a^2 - ab) = (a^2 - ab)$ | *Factor, collect terms* |
| $2 = 1$ | *Divide both sides by $(a^2 - ab)$* |

*So, what is wrong with the proof?*

# Proofs

# Proofs with Quantifiers

- Rules of inference can be extended in a straightforward manner to quantified statements
- **Universal Instantiation**: Given the premise that $\forall x P(x)$ and $c \in$ UoD (where UoD is the universe of discourse), we conclude that P(c) holds
- **Universal Generalization**: Here, we select an arbitrary element in the universe of discourse $c \in$ UoD and show that P(c) holds. We can therefore conclude that $\forall x P(x)$ holds
- **Existential Instantiation**: Given the premise that $\exists x P(x)$ holds, we simply give it a name, c, and conclude that P(c) holds
- **Existential Generalization:** Conversely, we establish that P(c) holds for a specific $c \in$ UoD, then we can conclude that $\exists x P(x)$

# Proofs with Quantifiers: Example (1)

- Show that "A car in the garage has an engine problem" and "Every car in the garage has been sold" imply the conclusion "A car that has been sold has an engine problem"
- Let
  - G(x): "x is in the garage"
  - E(x): "x has an engine problem"
  - S(x): "x has been sold"
- Let UoD be the set of all cars
- The premises are as follows:
  - $\exists x\,(G(x) \wedge E(x))$
  - $\forall x\,(G(x) \rightarrow S(x))$
- The conclusion we want to show is: $\exists x\,(S(x) \wedge E(x))$

# Proofs with Quantifiers: Example (2)

| | | |
|---|---|---|
| 1. | $\exists x\,(G(x) \wedge E(x))$ | *1st premise* |
| 2. | $(G(c) \wedge E(c))$ | *Existential instantiation of (1)* |
| 3. | $G(c)$ | *Simplification of (2)* |
| 4. | $\forall x\,(G(x) \rightarrow S(x))$ | *2nd premise* |
| 5. | $G(c) \rightarrow S(c)$ | *Universal instantiation of (4)* |
| 6. | $S(c)$ | *Modus ponens on (3) and (5)* |
| 7. | $E(c)$ | *Simplification from (2)* |
| 8. | $S(c) \wedge E(c)$ | *Conjunction of (6) and (7)* |
| 9. | $\exists x\,(S(x) \wedge E(x))$ | *Existential generalization of (8)* |

QED

# Trivial Proofs (1)

- Conclusion holds without using the premise
- A trivial proof can be given when the conclusion is shown to be (always) true.
- That is, if q is true, then $p \rightarrow q$ is true
- Examples
  - 'If CSE235 is easy implies that the Earth is round'
  - Prove 'If $x>0$ then $(x+1)^2 - 2x \geq x^2$'

# Trivial Proofs (2)

- Proof. It is easy to see:

  $(x+1)^2 - 2x$

  $= (x^2 + 2x + 1) - 2x$

  $= x^2 + 1$

  $\geq x^2$

- Note that the conclusion holds without using the hypothesis.

# Vacuous Proofs

- If the premise p is false
- Then the implication $p \rightarrow q$ is always true
- A vacuous proof is a proof that relies on the fact that no element in the universe of discourse satisfies the premise (thus the statement exists in vacuum in the UoD).
- Example:
  - If x is a prime number divisible by 16, then $x^2 < 0$
- No prime number is divisible by 16, thus this statement is true (counter-intuitive as it may be)

# Direct Proofs

- Most of the proofs we have seen so far are direct proofs
- In a direct proof
  - You assume the hypothesis p, and
  - Give a direct series (sequence) of implications
  - Using the rules of inference
  - As well as other results (proved independently)
  - To show that the conclusion q holds.

# Proof by Contrapositive (indirect proof)

- Recall that $(p \rightarrow q) \equiv (\neg q \rightarrow \neg p)$
- This is the basis for the proof by contraposition
  - You assume that the conclusion is false, then
  - Give a series of implications to show that
  - Such an assumption implies that the premise is false
- Example
  - Prove that if $x^3 < 0$ then $x < 0$

# Proof by Contrapositive: Example

- The contrapositive is "if $x \geq 0$ then $x^3 \geq 0$"
- Proof:
  1. If $x=0 \rightarrow x^3=0 \geq 0$
  2. If $x>0 \rightarrow x^2>0 \rightarrow x^3>0$ QED

# Proof by Contradiction

- To prove a statement p is true
  - you may assume that it is false
  - And then proceed to show that such an assumption leads a contradiction with a known result
- In terms of logic, you show that
  - for a known result r,
  - $(\neg p \rightarrow (r \wedge \neg r))$ is true
  - Which yields a contradiction c = $(r \wedge \neg r)$ cannot hold
- Example: $\sqrt{2}$ is an irrational number

# Proof by Contradiction: Example

- Let p be the proposition '$\sqrt{2}$ is an irrational number'
- Assume $\neg p$ holds, and show that it yields a contradiction
- $\sqrt{2}$ is rational

$\rightarrow \sqrt{2} = a/b$, a, b $\in R$ and a, b have no common factor (proposition r) — *Definition of rational numbers*

$\rightarrow 2 = a^2/b^2$ — *Squarring the equation*

$\rightarrow (2b^2 = a^2) \rightarrow (a^2$ is even$) \rightarrow (a = 2c)$ — *Algebra*

$\rightarrow (2b^2 = 4c^2) \rightarrow (b^2 = 2c^2) \rightarrow (b^2$ is even$) \rightarrow$ (b is even) — *Algebra*

$\rightarrow$ (a, b are even) $\rightarrow$ (a, b have a common factor 2) $\rightarrow \neg r$

$\rightarrow (\neg p \rightarrow (r \wedge \neg r))$, which is a contradiction

So, ($\neg p$ is false) $\rightarrow$ (p is true), which means $\sqrt{2}$ is irrational

# Proof by Cases

- Sometimes it is easier to prove a theorem by
  - Breaking it down into cases and
  - Proving each one separately
- Example:
  - Let $n \in Z$. Prove that $9n^2+3n-2$ is even

# Proof by Cases: Example

- Observe that $9n^2+3n-2=(3n+2)(3n-1)$
- n is an integer $\rightarrow$(3n+2)(3n-1) is the product of two integers
- **Case 1**: Assume 3n+2 is even

  $\rightarrow$ $9n^2+3n-2$ is trivially even because it is the product of two integers, one of which is even
- **Case 2**: Assume 3n+2 is odd

  $\rightarrow$ 3n+2-3 is even $\rightarrow$ 3n-1 is even $\rightarrow$ $9n^2+3n-2$ is even because one of its factors is even □

# Existence Proofs

- A **constructive existence proof** asserts a theorem by providing a specific, concrete example of a statement
  - Such a proof only proves a statement of the form $\exists x P(x)$ for some predicate P.
  - It does not prove the statement for all such x
- A **nonconstructive existence proof** also shows a statement of the form $\exists x P(x)$, but is does not necessarily need to give a specific example x.
  - Such a proof usually proceeds by contradiction:
    - Assume that $\neg\exists x P(x) \equiv \forall x \neg P(x)$ holds
    - Then get a contradiction

# Uniqueness Proofs

- A **uniqueness proof** is used to show that a certain element (specific or not) has a certain property.
- Such a proof usually has two parts
  1. A proof of existence: $\exists x P(x)$
  2. A proof of uniqueness: if $x \neq y$ then $\neg P(y)$)
- Together we have the following:

$$\exists x \, ( \, P(x) \wedge (\forall y \, (x \neq y \rightarrow \neg P(y) \, ) \, )$$

# Counter Examples

- Sometimes you are asked to disprove a statement
- In such a situation you are actually trying to prove the negation of the statement
- With statements of the form $\forall x\, P(x)$, it suffices to give a counter example
  - because the existence of an element x for which $\neg P(x)$ holds proves that $\exists x\, \neg P(x)$
  - which is the negation of $\forall x\, P(x)$

# Counter Examples: Example

- Example: Disprove $n^2+n+1$ is a prime number for all $n \geq 1$
- A simple counterexample is n=4.
- In fact: for n=4, we have

$n^2+n+1 = 4^2+4+1$

$= 16+4+1$

$= 21 = 3 \times 7$, which is clearly not prime QED

# Counter Examples: A Word of Caution

- No matter how many examples you give, you can never prove a theorem by giving examples (unless the universe of discourse is finite—why?—which is in called an exhaustive proof)
- Counter examples can only be used to disprove universally quantified statements
- Do not give a proof by simply giving an example

# Proof Strategies

- Example: Forward and backward reasoning
- If there were a single strategy that always worked for proofs, mathematics would be easy
- The best advice we can give you:
  - Beware of fallacies and circular arguments (i.e., begging the question)
  - Don't take things for granted. Try proving assertions first before you can take/use them as facts
  - Don't peek at proofs. Try proving something for yourself before looking at the proof
  - If you peeked, challenge yourself to reproduce the proof later on.. w/o peeking again
  - The best way to improve your proof skills is PRACTICE.

# Applications

# Translating English Sentences

- Steps to convert an English sentence to a statement in propositional logic
- Identify atomic propositions and represent using propositional variables.
- Determine appropriate logical connectives
- "If I go to Harry's or to the country, I will not go shopping."
- p: I go to Harry's
- q: I go to the country.
- r: I will go shopping.

If If p or q then not r.
$(p \wedge q) \rightarrow \neg r$.

# Example

- Problem: Translate the following sentence into propositional logic:
- “You can access the Internet from campus only if you are a computer science major or you are not a freshman.”
- One Solution: Let a, c, and f represent respectively “You can access the internet from campus,” “You are a computer science major,” and “You are a freshman.”

- $a \rightarrow (c \vee \neg f)$

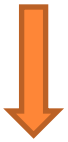
# Boolean Searches

- Logical connectives are used extensively in searches of large collections of information, such as indexes of Web pages. In Boolean searches,

| Connective | |
|---|---|
| AND | match records that contain both of two search terms |
| OR | match one or both of two search terms |
| NOT | exclude a particular search term. (sometimes written as AND NOT ) |

# Boolean Searches

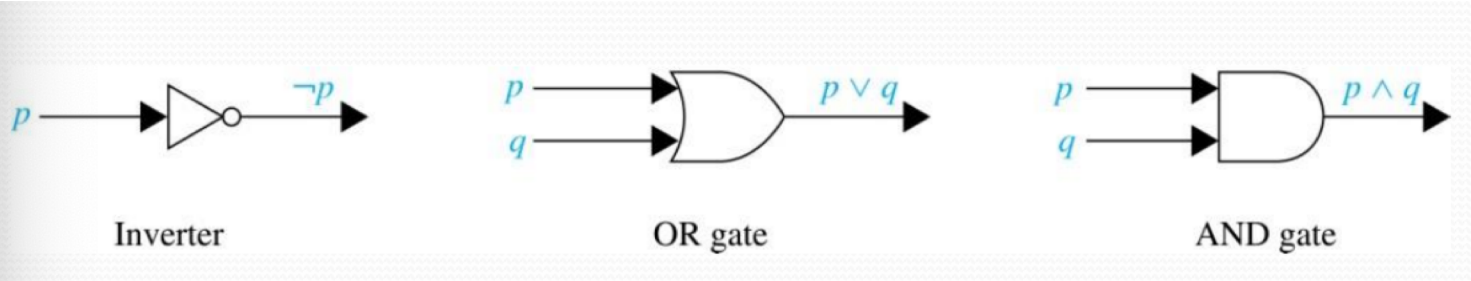
- Web Page Searching Most Web search engines support Boolean searching techniques, which usually can help find Web pages about particular subjects.
- Example,
- to find Web pages about universities in New Mexico



- NEW AND MEXICO AND UNIVERSITIES

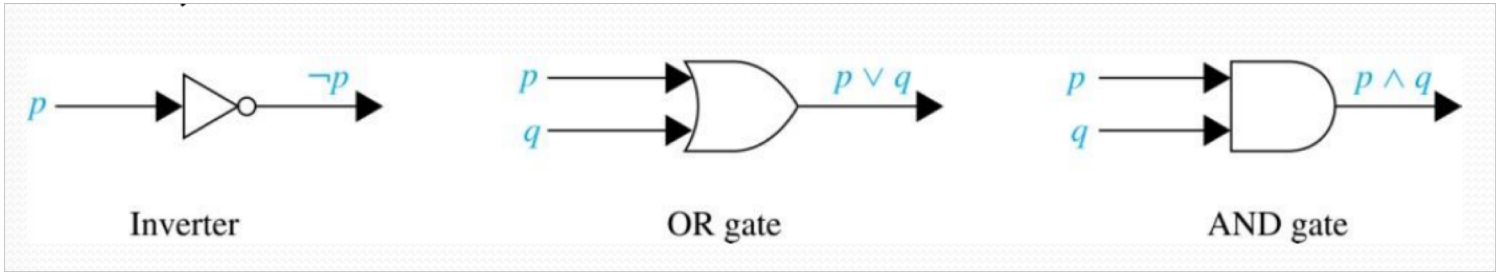
# Logic Circuits

- Electronic circuits; each input/output signal can be viewed as a 0 or 1.
  - 0 represents False
  - 1 represents True
  - Complicated circuits are constructed from three basic circuits called gates.



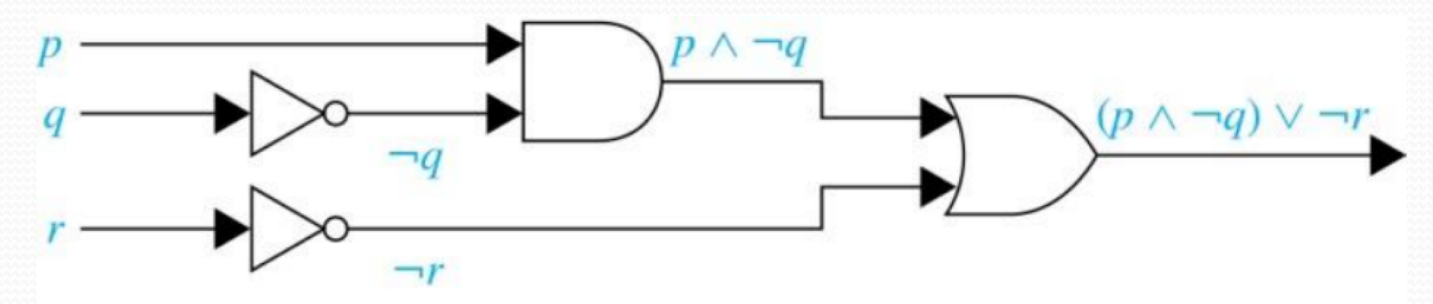
# Logic Circuits

- The inverter (NOT gate) takes an input bit and produces the negation of that bit.
- The OR gate takes two input bits and produces the value equivalent to the disjunction of the two bits.
- The AND gate takes two input bits and produces the value equivalent to the conjunction of the two bits.



# Logic Circuits

- More complicated digital circuits can be constructed by combining these basic circuits to produce the desired output given the input signals by building a circuit for each piece of the output expression and then combining them. For example:



# Next class

- Topic: Introduction I and II
- Pre-class reading: Chap 5