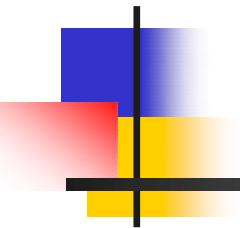
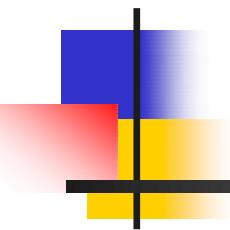


Discrete Mathematics and Applications



Dept. Information Technology,
CBIT

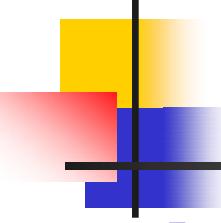


Lecture 1

Course Overview

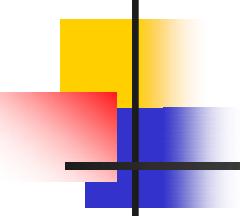
Chapter 1. The Foundations

1.1 Propositional Logic



Resources

- Textbook: *Discrete Mathematics and Its Applications (6th Edition)*, by Kenneth H. Rosen, McGraw-Hill
- Web site:
 - 1. https://onlinecourses.nptel.ac.in/noc18_cs53/
 - 2. <https://www.coursera.org/learn/discrete-mathematics>

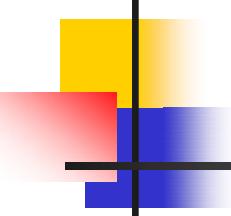


What is Mathematics, really?

- It's *not* just about numbers!
- Mathematics is *much* more than that:

Mathematics is, most generally, the study of
any and all *absolutely certain* truths about
any and all *perfectly well-defined* concepts.

- These concepts can be *about* numbers, symbols, objects, images, sounds, *anything*!
- It is a way to interpret the world around you.



So, what's *this* class about?

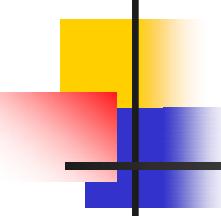
What are “discrete structures” anyway?

- “**Discrete**” - Composed of distinct, separable parts. (Opposite of *continuous*.)

discrete:continuous :: digital:analog

- “**Structures**” - Objects built up from simpler objects according to some definite pattern.

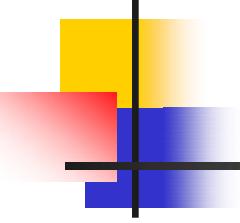
- “**Discrete Mathematics**” - The study of discrete, mathematical (i.e. well-defined conceptual) objects and structures.



Discrete Objects/Concepts and Structures We Study

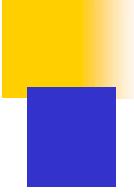
- DM PART I
 - Propositions
 - Predicates
 - Proofs
 - Sets
 - Functions
 - Algorithms
 - Integers
 - Summations
 - Sequences
 - Strings
 - Permutations
 - Combinations
 - Probability

- DM PART II
 - Relations
 - Graphs
 - Trees
 - Boolean Functions / Logic Circuits



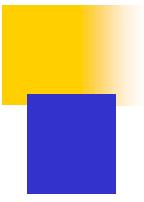
Why to Study Discrete Math?

- The basis of all of digital information processing is: *Discrete manipulations of discrete structures represented in memory.*
- It's the basic language and conceptual foundation for all of computer science.
- Discrete math concepts are also widely used throughout math, science, engineering, economics, biology, etc., ...



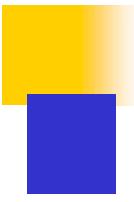
Why to Study Discrete Math?

- To learn a set of Mathematical facts and how to apply them.
- How to think logically and mathematically.
- A generally useful tool for rational thought!
- Discrete Mathematics: Is a part of mathematics to study of discrete objects.



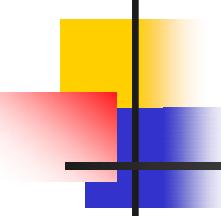
Problems solved

- How many ways are there to choose a valid password on a computer system?
- What is the probability of winning a lottery?
- Is there a link between two computers in a network?
- How can I identify spam e-mail messages?
- How can I encrypt a message so that no unintended recipient can read it?
- What is the shortest path between two cities using a transportation system?



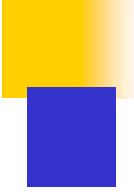
Uses of Discrete Math

- Discrete Mathematics is used whenever objects are counted, when relationships between finite sets are studied.
- Develops ability to understand and create mathematical arguments.
- Discrete mathematics is the gateway to more advanced courses in all parts of the mathematical sciences.



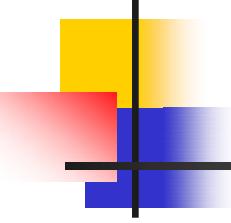
Uses for Discrete Math in Computer Science

- Advanced algorithms & data structures
- Programming language compilers & interpreters
- Computer networks
- Operating systems
- Computer architecture
- Database management systems
- Cryptography
- Error correction codes
- Graphics & animation algorithms, game engines,
etc....
- *i.e.*, the whole field!



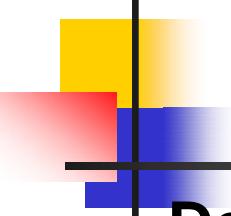
LOGIC

- Rules of logic specify meaning of mathematical statements.
- logic has practical applications to design computing machines.
- Proof –makes a correct mathematical argument.
- Theorem- mathematical statement is True.
- To learn a mathematical topic, we need to construct mathematical argument on that topic.
- Proofs are used to verify that computer programs and algorithms produce correct result.



1.1 Propositional Logic

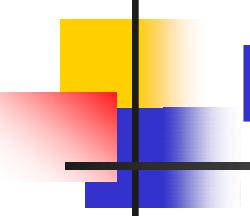
- Logic
 - Study of reasoning.
 - Specifically concerned with whether reasoning is correct.
 - Focuses on the relationship among statements, not on the content of any particular statement.
 - Gives precise meaning to mathematical statements.
- ***Propositional Logic*** is the logic that deals with statements (propositions) and compound statements built from simpler statements using so-called *Boolean connectives*.
- Some applications in computer science:
 - Design of digital electronic circuits.
 - Expressing conditions in programs.
 - Queries to databases & search engines.



Definition of a *Proposition*

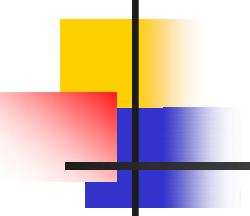
Definition: A ***proposition*** (denoted p, q, r, \dots) is simply:

- a *statement* (i.e., a declarative sentence which declares a fact)
 - *with some definite meaning*, (not vague or ambiguous)
- having a *truth value* that's either *true (T)* or *false (F)*
 - it is **never** both, neither, or somewhere “in between!”
 - However, you might not *know* the actual truth value,
 - and, the truth value might *depend* on the situation or context.
- Later, we will study *probability theory*, in which we assign *degrees of certainty* (“between” **T** and **F**) to propositions.
 - But for now: think True/False only! (or in terms of **1** and **0**)



Examples of Propositions

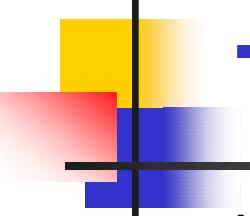
- It is raining. (In a given situation)
- Beijing is the capital of China. (T)
- $2 + 2 = 5$. (F)
- $1 + 2 = 3$. (T)
- A fact-based declaration is a proposition, even if no one knows whether it is true
 - 11213 is prime.
 - There exists an odd perfect number.



Examples of Non-Proposition

The following are **NOT** propositions:

- Who's there? (interrogative, question)
- Just do it! (imperative, command)
- jhhggggg. (meaningless interjection)
- Yeah, I sorta dunno, whatever... (vague)
- $1 + 2$ (expression with a non-true/false value)
- $x + 2 = 5$ (declaration about semantic tokens of non-constant value)

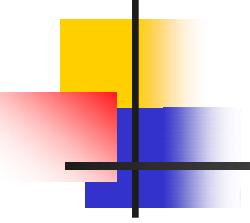


Truth Tables

- An *operator* or *connective* combines one or more *operand* expressions into a larger expression. (e.g., “+” in numeric expressions.)
- **Unary** operators take *one* operand (e.g., -3); **Binary** operators take *two* operands (e.g. 3×4).
- **Propositional** or **Boolean operators** operate on propositions (or their truth values) instead of on numbers.
- The **Boolean domain** is the set $\{T, F\}$. Either of its elements is called a **Boolean value**.

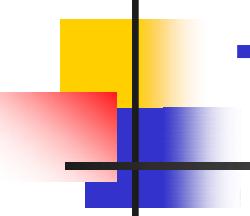
An n -tuple (p_1, \dots, p_n) of Boolean values is called a **Boolean n -tuple**.

- An n -operand truth table is a table that assigns a Boolean value to the set of all Boolean n -tuples.



Some Popular Boolean Operators

<u>Formal Name</u>	<u>Nickname</u>	<u>Arity</u>	<u>Symbol</u>
Negation operator	NOT	Unary	\neg
Conjunction operator	AND	Binary	\wedge
Disjunction operator	OR	Binary	\vee
Exclusive-OR operator	XOR	Binary	\oplus
Implication operator	IMPLIES	Binary	\rightarrow
Biconditional operator	IFF	Binary	\leftrightarrow



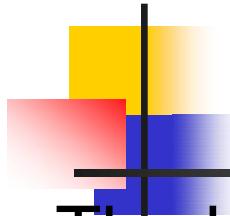
The Negation Operator

- The unary ***negation operator*** “ \neg ” (*NOT*) transforms a proposition into its logical *negation*.
- *E.g.* If p = “I have brown hair.”
then $\neg p$ = “It is not the case that I have brown hair” or “I do **not** have brown hair.”
- The *truth table* for NOT:

p	$\neg p$
T	F
F	T

Operand
column

Result
column



The Conjunction Operator

- The binary ***conjunction operator*** “ \wedge ” (AND) combines two propositions to form their logical *conjunction*.

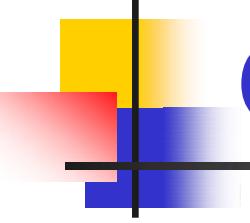
E.g. If p = “Rebecca’s PC has more than 16 GB free hard disk space.”

and

q = “The processor in Rebecca’s PC runs faster than 1 GHz.”

then, $p \wedge q$ = “Rebecca’s PC has more than 16 GB free hard disk space, and the processor in Rebecca’s PC runs faster than 1 GHz.”

“Rebecca’s PC has more than 16 GB free hard disk space, and its processor runs faster than 1 GHz.”

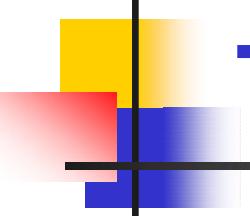


Conjunction Truth Table

Operand columns

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

- Note that a conjunction $p_1 \wedge p_2 \wedge \dots \wedge p_n$ of n propositions will have 2^n rows in its truth table

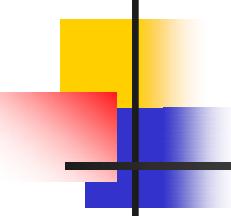


The Disjunction Operator

- The binary ***disjunction operator*** “ \vee ” (*OR*) combines two propositions to form their logical *disjunction*.
- *E.g.* If p = “My car has a bad engine.” and q = “My car has a bad carburetor.”

then, $p \vee q$ = “My car has a bad engine, **or** my car has a bad carburetor.”

Meaning is like “and/or” in informal English.

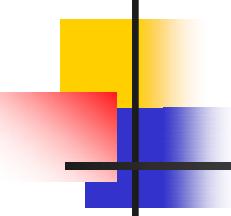


Disjunction Truth Table

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

Note difference
from AND

- Note that $p \vee q$ means that p is true, or q is true, **or both** are true!
- So, this operation is also called ***inclusive or***, because it **includes** the possibility that both p and q are true.



The Exclusive-Or Operator

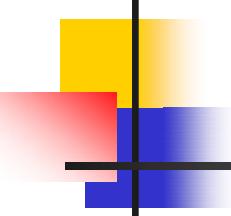
- The binary **exclusive-or operator** “ \oplus ” (*XOR*) combines two propositions to form their logical “exclusive or”
- *E.g.* If p = “I will earn an A in this course.” and q = “I will drop this course.”, then
 $p \oplus q$ = “I will **either** earn an A in this course, **or** I will drop it (**but not both!**)”

Exclusive-Or Truth Table

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

Note difference
from OR.

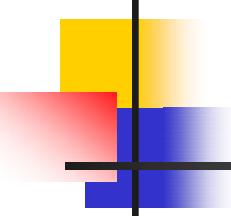
- Note that $p \oplus q$ means that p is true, or q is true, but **not both!**
- This operation is called **exclusive or**, because it **excludes** the possibility that both p and q are true.



Natural Language is Ambiguous

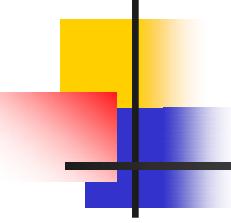
- Note that the English “or” can be ambiguous regarding the “both” case!
- “Pat is a singer or Pat is a writer.” - \vee
- “Pat is a man or Pat is a woman.” - \oplus
- Need context to disambiguate the meaning!
- For this class, assume “or” means inclusive (\vee).

p	q	p “or” q
T	T	?
T	F	T
F	T	T
F	F	F



The Implication Operator

- The conditional statement (aka *implication*)
 $p \rightarrow q$ states that p implies q .
- *I.e., If p is true, then q is true; but if p is not true, then q could be either true or false.*
- *E.g., let p = “You study hard.”
 q = “You will get a good grade.”
 $p \rightarrow q$ = “If you study hard, then you will get a good grade.” (else, it could go either way)*
 - p : *hypothesis* or *antecedent* or *premise*
 - q : *conclusion* or *consequence*



Implication Truth Table

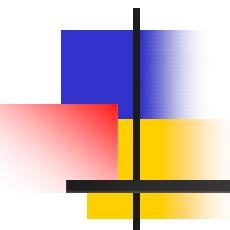
p	q	$p \rightarrow q$
T	T	T
T	F	F }
F	T	T
F	F	T

The only
False case!

- $p \rightarrow q$ is **false only** when p is true but q is **not** true.
- $p \rightarrow q$ does **not** require that p or q **are ever true!**
- *E.g.* “ $(1=0) \rightarrow$ pigs can fly” is TRUE!

Discrete Mathematics for Computer Science

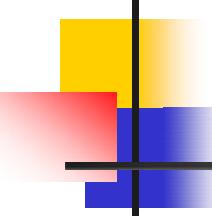




Lecture 2

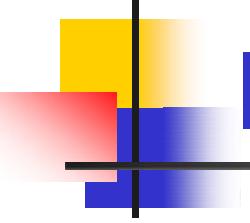
Chapter 1. The Foundations

1.1 Propositional Logic



Review: The Implication Operator

- The conditional statement (*implication*)
 $p \rightarrow q$ states that p implies q .
- I.e., If p is true, then q is true; but if p is not true, then q could be either true or false.
- E.g., let p = “You study hard.”
 q = “You will get a good grade.”
 $p \rightarrow q$ = “If you study hard, then you will get a good grade.” (else, it could go either way)
 - p : *hypothesis* or *antecedent* or *premise*
 - q : *conclusion* or *consequence*

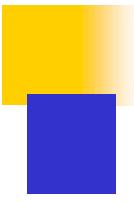


Review: Implication Truth Table

p	q	$p \rightarrow q$
T	T	T
T	F	F }
F	T	T
F	F	T

The only
False case!

- $p \rightarrow q$ is **false only** when p is true but q is **not** true.
- $p \rightarrow q$ does **not** require that p or q **are ever true!**
 - E.g. “ $(1=0) \rightarrow$ pigs can fly” is TRUE!



$p \rightarrow q$ English Phrases

- “ p implies q ”
- “if p , then q ”
- “if p, q ”
- “when p, q ”
- “whenever p, q ”
- “ q if p ”
- “ q when p ”
- “ q whenever p ”
- “ p only if q ”
- “ p is sufficient for q ”
- “ q is necessary for p ”
- “ q follows from p ”
- “ q is implied by p ”

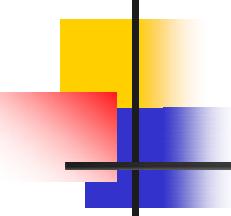
Converse, Inverse, Contrapositive

- Some terminology, for an implication $p \rightarrow q$:
- Its **converse** is: $q \rightarrow p$.
- Its **inverse** is: $\neg p \rightarrow \neg q$.
- Its **contrapositive**: $\neg q \rightarrow \neg p$.

p	q	$p \rightarrow q$	$q \rightarrow p$	$\neg p \rightarrow \neg q$	$\neg q \rightarrow \neg p$
T	T	T	T	T	T
T	F	F	T	T	F
F	T	T	F	F	T
F	F	T	T	T	T

- One of these three has the *same meaning* (same truth table) as $p \rightarrow q$. Can you figure out which?

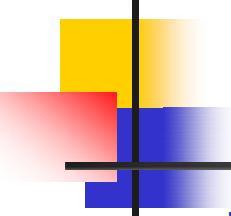
Contrapositive



Examples

- p : Today is Easter
- q : Tomorrow is Monday

- $p \rightarrow q$:
 If today is Easter then tomorrow is Monday.
- **Converse**: $q \rightarrow p$
 If tomorrow is Monday then today is Easter.
- **Inverse**: $\neg p \rightarrow \neg q$
 If today is not Easter then tomorrow is not Monday.
- **Contrapositive**: $\neg q \rightarrow \neg p$
 If tomorrow is not Monday then today is not Easter.

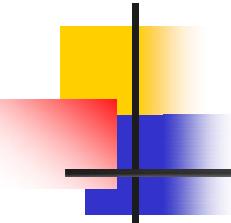


The Biconditional Operator

- The **biconditional** statement $p \leftrightarrow q$ states that p **if and only if** (iff) q .
- p = “It is below freezing.”
 q = “It is snowing.”
 $p \leftrightarrow q$ = “It is below freezing if and only if it is snowing.”

or

= “That it is below freezing is necessary and sufficient for it to be snowing”

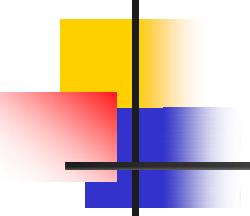


Biconditional Truth Table

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

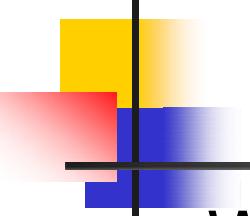
- p is necessary and sufficient for q
- If p then q , and conversely
- p iff q

- $p \leftrightarrow q$ is equivalent to $(p \rightarrow q) \wedge (q \rightarrow p)$.
- $p \leftrightarrow q$ means that p and q have the **same** truth value.
- $p \leftrightarrow q$ does **not** imply that p and q are true.
- Note this truth table is the exact **opposite** of \oplus 's! Thus, $p \leftrightarrow q$ means $\neg(p \oplus q)$.



Boolean Operations

- Conjunction: $p \wedge q$, (read p and q), “discrete math is a required course **and** I am a computer science major”.
- Disjunction: , $p \vee q$, (read p or q), “discrete math is a required course **or** I am a computer science major”.
- Exclusive or: $p \oplus q$, “discrete math is a required course **or** I am a computer science major **but not both**”.
- Implication: $p \rightarrow q$, “**if** discrete math is a required course **then** I am a computer science major”.
- Biconditional: $p \leftrightarrow q$, “discrete math is a required course **if and only if** I am a computer science major”.

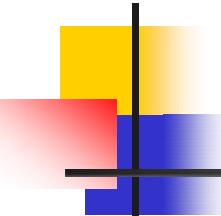


Boolean Operations

- We have seen 1 unary operator and 5 binary operators. What are they? Their truth tables are below.

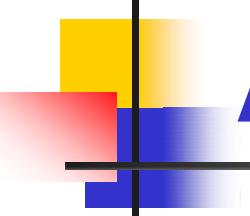
p	q	$\neg p$	$p \wedge q$	$p \vee q$	$p \oplus q$	$p \rightarrow q$	$p \leftrightarrow q$
T	T	F	T	T	F	T	T
T	F	F	F	T	T	F	F
F	T	T	F	T	T	T	F
F	F	T	F	F	F	T	T

- For an implication $p \rightarrow q$
- Its **converse** is: $q \rightarrow p$
- Its **inverse** is: $\neg p \rightarrow \neg q$
- Its **contrapositive**: $\neg q \rightarrow \neg p$



Compound Propositions

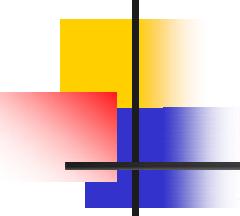
- A ***propositional variable*** is a variable such as p , q , r (possibly subscripted, e.g. p_j) over the Boolean domain.
- An ***atomic proposition*** is either Boolean constant or a propositional variable: e.g. T , F , p
- A ***compound proposition*** is derived from atomic propositions by application of propositional operators:
e.g. $\neg p$, $p \vee q$, $(p \vee \neg q) \rightarrow q$
- Precedence of logical operators: \neg , \wedge , \vee , \rightarrow , \leftrightarrow
- Precedence also can be indicated by parentheses.
 - e.g. $\neg p \wedge q$ means $(\neg p) \wedge q$, not $\neg(p \wedge q)$



An Exercise

- Any compound proposition can be evaluated by a truth table
- $(p \vee \neg q) \rightarrow q$

p	q	$\neg q$	$p \vee \neg q$	$(p \vee \neg q) \rightarrow q$
T	T	F	T	T
T	F	T	T	F
F	T	F	F	T
F	F	T	T	F



Translating English Sentence

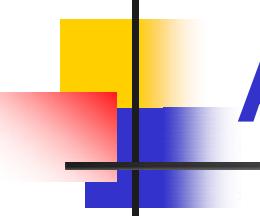
- Let p = “It rained last night”,
 q = “The sprinklers came on last night,”
 r = “The lawn was wet this morning.”

Translate each of the following into English:

$$\neg p = \text{“It didn’t rain last night.”}$$

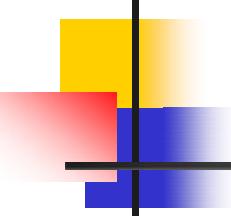
$$r \wedge \neg p = \text{“The lawn was wet this morning, and it didn’t rain last night.”}$$

$$\neg r \vee p \vee q = \text{“The lawn wasn’t wet this morning, or it rained last night, or the sprinklers came on last night.”}$$



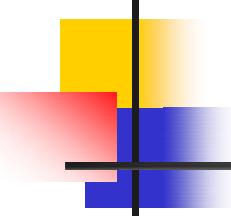
Another Example

- Find the converse of the following statement.
 - “Raining tomorrow is a sufficient condition for my not going to town.”
- **Step 1:** Assign propositional variables to component propositions.
 - p : It will rain tomorrow
 - q : I will not go to town
- **Step 2:** Symbolize the assertion: $p \rightarrow q$
- **Step 3:** Symbolize the converse: $q \rightarrow p$
- **Step 4:** Convert the symbols back into words.
 - “If I don’t go to town then it will rain tomorrow” or
 - “Raining tomorrow is a *necessary condition* for my not going to town.”



Logic and Bit Operations

- A ***bit*** is a **binary** (base 2) **digit**: 0 or 1.
- Bits may be used to represent truth values.
 - By convention:
0 represents “**False**”; 1 represents “**True**”.
- A ***bit string of length n*** is an ordered sequence of $n \geq 0$ bits.
- By convention, bit strings are (sometimes) written left to right:
 - e.g. the “first” bit of the bit string “1001101010” is 1.
 - What is the length of the above bit string?



Bitwise Operations

- Boolean operations can be extended to operate on bit strings as well as single bits.
- Example:

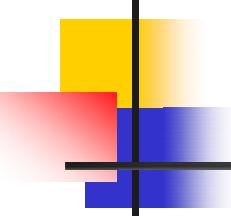
01 1011 0110

11 0001 1101

11 1011 1111 Bit-wise OR

01 0001 0100 Bit-wise AND

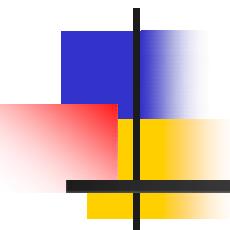
10 1010 1011 Bit-wise XOR



End of 1.1

You have learned about:

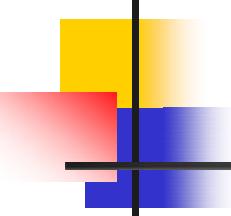
- Propositions: what they are
- Propositional logic operators'
 - symbolic notations, truth tables, English equivalents, logical meaning
- Atomic vs. compound propositions
- Bits, bit strings, and bit operations
- Next section:
 - Propositional equivalences
 - Equivalence laws
 - Proving propositional equivalences



Lecture 3

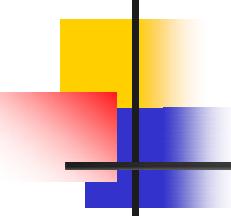
Chapter 1. The Foundations

- 2. Propositional Equivalences
- 3. Predicates and Quantifiers



1.2 Propositional Equivalence

- A **tautology** is a compound proposition that is **true** *no matter what* the truth values of its atomic propositions are!
 - e.g. $p \vee \neg p$ (“Today the sun will shine or today the sun will not shine.”) [What is its truth table?]
- A **contradiction** is a compound proposition that is **False**.
 - e.g. $p \wedge \neg p$ (“Today is Wednesday and today is not Wednesday.”) [Truth table?]
- A **contingency** is a compound proposition that is neither a tautology nor a contradiction.
 - e.g. $(p \vee q) \rightarrow \neg r$



Logical Equivalence

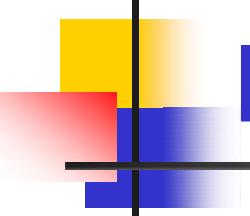
- Compound proposition p is ***logically equivalent*** to compound proposition q , written $p \equiv q$ or $p \Leftrightarrow q$, **iff** the compound proposition $p \leftrightarrow q$ is a tautology.
- Compound propositions p and q are logically equivalent to each other **iff** p and q contain the same truth values as each other in all corresponding rows of their truth tables.

Proving Equivalence via Truth Tables

- Prove that $\neg(p \wedge q) \equiv \neg p \vee \neg q$. (De Morgan's law)

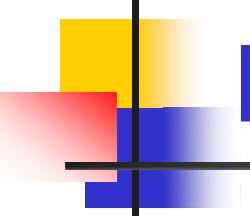
p	q	$p \wedge q$	$\neg p$	$\neg q$	$\neg p \vee \neg q$	$\neg(p \wedge q)$
T	T	T	F	F	F	F
T	F	F	F	T	T	T
F	T	F	T	F	T	T
F	F	F	T	T	T	T

- Show that Check out the solution in the textbook!
- $\neg(p \vee q) \equiv \neg p \wedge \neg q$ (De Morgan's law)
- $p \rightarrow q \equiv \neg p \vee q$
- $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ (distributive law)



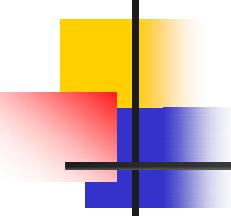
Equivalence Laws

- These are similar to the arithmetic identities you may have learned in algebra, but for propositional equivalences instead.
- They provide a pattern or template that can be used to match part of a much more complicated proposition and to find an equivalence for it and possibly simplify it.



Equivalence Laws

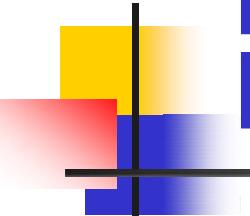
- *Identity:* $p \wedge T \equiv p$ $p \vee F \equiv p$
- *Domination:* $p \vee T \equiv T$ $p \wedge F \equiv F$
- *Idempotent:* $p \vee p \equiv p$ $p \wedge p \equiv p$
- *Double negation:* $\neg\neg p \equiv p$
- *Commutative:* $p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$
- *Associative:* $(p \vee q) \vee r \equiv p \vee (q \vee r)$
 $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$



More Equivalence Laws

- *Distributive:* $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$
 $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$
- *De Morgan's:*
 $\neg(p \wedge q) \equiv \neg p \vee \neg q$
 $\neg(p \vee q) \equiv \neg p \wedge \neg q$
- *Absorption*
 $p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$
- *Trivial tautology/contradiction:*
 $p \vee \neg p \equiv T$ $p \wedge \neg p \equiv F$

See Table 6, 7, and 8 of Section 1.2



Defining Operators via Equivalences

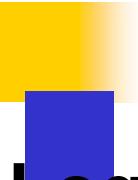
Using equivalences, we can *define* operators in terms of other operators.

Exclusive or: $p \oplus q \equiv (p \wedge \neg q) \vee (\neg p \wedge q)$

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

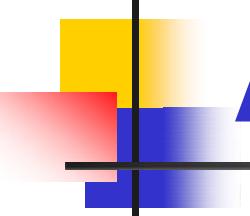
- Implies: $p \rightarrow q \equiv \neg p \vee q$
- Biconditional: $p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$
 $p \leftrightarrow q \equiv \neg(p \oplus q)$
 $p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$
 $p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$
 $\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$

This way we can “normalize” propositions



Logical Equivalences Involving Conditional Statements.

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



An Example Problem

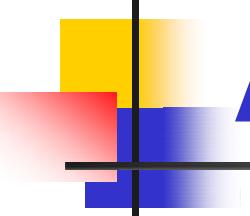
- Show that $\neg(p \rightarrow q)$ and $p \wedge \neg q$ are logically equivalent.

$$\neg(p \rightarrow q) \quad [\text{Expand definition of } \rightarrow]$$

$$\equiv \neg(\neg p \vee q) \quad [\text{DeMorgan's Law}]$$

$$\equiv \neg(\neg p) \wedge \neg q \quad [\text{Double Negation}]$$

$$\equiv p \wedge \neg q$$



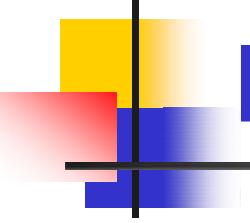
Another Example Problem

- Check using a symbolic derivation whether

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

$$\begin{aligned}(p \wedge \neg q) &\rightarrow (p \oplus r) \quad [\text{Expand definition of } \rightarrow] \\&\equiv \neg(p \wedge \neg q) \vee (p \oplus r) \quad [\text{Expand definition of } \oplus] \\&\equiv \neg(p \wedge \neg q) \vee ((p \vee r) \wedge \neg(p \wedge r)) \\&\qquad\qquad\qquad [\text{DeMorgan's Law}] \\&\equiv (\neg p \vee q) \vee ((p \vee r) \wedge \neg(p \wedge r))\end{aligned}$$

cont.

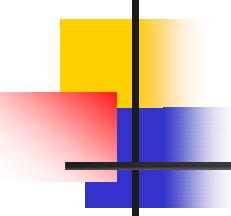


Example Continued...

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

$$\begin{aligned} & (\neg p \vee q) \vee ((p \vee r) \wedge \neg(p \wedge r)) \quad [\vee \text{ Commutative}] \\ & \equiv (\neg p \vee q) \vee ((p \vee r) \wedge \neg(p \wedge r)) \quad [\vee \text{ Associative}] \\ & \equiv q \vee (\neg p \vee ((p \vee r) \wedge \neg(p \wedge r))) \quad [\text{Distribute } \vee \text{ over } \wedge] \\ & \equiv q \vee ((\neg p \vee (p \vee r)) \wedge (\neg p \vee \neg(p \wedge r))) \quad [\vee \text{ Assoc.}] \\ & \equiv q \vee ((\neg p \vee p) \vee r) \wedge (\neg p \vee \neg(p \wedge r)) \quad [\text{Trivial taut.}] \\ & \equiv q \vee (\top \wedge (\neg p \vee \neg(p \wedge r))) \quad [\text{Domination}] \\ & \equiv q \vee (\top \wedge (\neg p \vee \neg(p \wedge r))) \quad [\text{Identity}] \\ & \equiv q \vee (\neg p \vee \neg(p \wedge r)) \end{aligned}$$

cont.



End of Long Example

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

$$q \vee (\neg p \vee \neg(p \wedge r)) \quad [\text{DeMorgan's Law}]$$

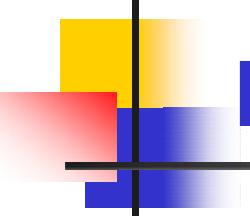
$$\equiv q \vee (\neg p \vee (\neg p \vee \neg r)) \quad [\vee \text{ Associative}]$$

$$\equiv q \vee ((\neg p \vee \neg p) \vee \neg r) \quad [\text{Idempotent}]$$

$$\equiv q \vee (\neg p \vee \neg r) \quad [\text{Associative}]$$

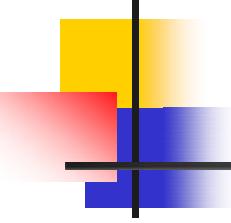
$$\equiv (q \vee \neg p) \vee \neg r \quad [\vee \text{ Commutative}]$$

$$\equiv \neg p \vee q \vee \neg r \blacksquare$$



Review: Propositional Logic

- Atomic propositions: p, q, r, \dots
- Boolean operators: $\neg \wedge \vee \oplus \rightarrow \leftrightarrow$
- Compound propositions: $(p \wedge \neg q) \vee r$
- Equivalences: $p \wedge \neg q \Leftrightarrow \equiv \neg(p \rightarrow q)$
- Proving equivalences using:
 - Truth tables
 - Symbolic derivations (series of logical equivalences) $p \equiv q \equiv r \equiv \dots$



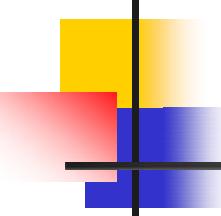
1.3 Predicate Logic

- Consider the sentence

“For every x , $x > 0$ ”

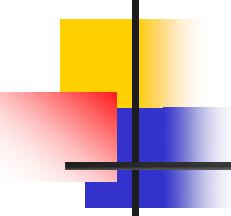
If this were a true statement about the positive integers, it could not be adequately symbolized using only statement letters, parentheses and logical connectives.

*The sentence contains two new features: a **predicate** and a **quantifier***



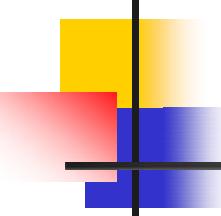
Subjects and Predicates

- In the sentence “The dog is sleeping”:
 - The phrase “the dog” denotes the **subject** – the *object* or *entity* that the sentence is about.
 - The phrase “is sleeping” denotes the **predicate** – a property that the subject of the statement can have.
- In predicate logic, a **predicate** is modeled as a ***propositional function* $P(\cdot)$** from subjects to propositions.
 - $P(x) = \text{“}x \text{ is sleeping}\text{”}$ (where x is any subject).
 - $P(\text{The cat}) = \text{“}The \ cat \text{ is sleeping}\text{”}$ (proposition!)



More About Predicates

- Convention: Lowercase variables $x, y, z\dots$ denote subjects; uppercase variables $P, Q, R\dots$ denote propositional functions (or predicates).
- Keep in mind that *the result of applying a predicate P to a value of subject x is the proposition*. But the predicate P , or the statement $P(x)$ **itself** (e.g. $P =$ “is sleeping” or $P(x) =$ “ x is sleeping”) is **not** a proposition.
 - e.g. if $P(x) =$ “ x is a prime number”, $P(3)$ is the *proposition* “3 is a prime number.”



Propositional Functions

- Predicate logic *generalizes* the grammatical notion of a predicate to also include propositional functions of **any** number of arguments, each of which may take **any** grammatical role that a noun can take.
 - e.g.:

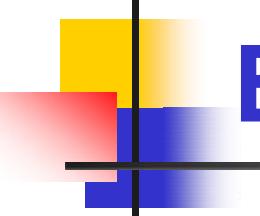
let $P(x,y,z)$ = “ x gave y the grade z ”

then if

x = “Mike”, y = “Mary”, z = “A”,

then

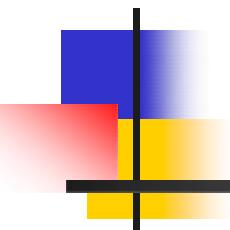
$P(x,y,z)$ = “**Mike** gave **Mary** the grade **A**.”



Examples

- Let $P(x)$: $x > 3$. Then
 - $P(4)$ is TRUE/FALSE
 - $P(2)$ is TRUE/FALSE
- Let $Q(x, y)$: x is the capital of y . Then
 - $Q(\text{Washington D.C.}, \text{U.S.A.})$ is TRUE
 - $Q(\text{Hilo}, \text{Hawaii})$ is FALSE
 - $Q(\text{Massachusetts}, \text{Boston})$ is FALSE
 - $Q(\text{Denver}, \text{Colorado})$ is TRUE
 - $Q(\text{New York}, \text{New York})$ is FALSE
- Read EXAMPLE 6
 - If $x > 0$ then $x := x + 1$ (in a computer program)

$4 > 3$
$2 > 3$



Lecture 4

Chapter 1. The Foundations

1.3 Introduction of Logic
Circuits, Predicates and
Quantifiers

Logic Circuits

- A logic circuit (or digital circuit) receives input signals p_1, p_2, \dots, p_n , each a bit [either 0 (off) or 1 (on)], and produces output signals s_1, s_2, \dots, s_n , each a bit.

1.2 Applications of Propositional Logic

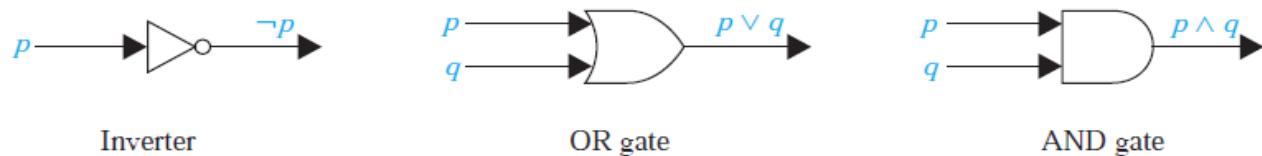


FIGURE 1 Basic logic gates.

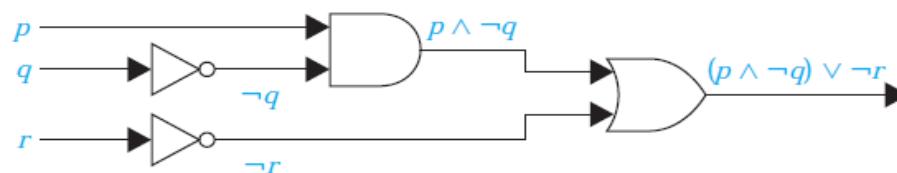
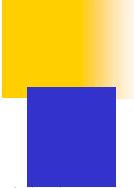


FIGURE 2 A combinatorial circuit.



EXAMPLE

Build a digital circuit that produces the output $(p \vee \neg r) \wedge (\neg p \vee (q \vee \neg r))$ when given input bits p, q, and r.

Solution:

To construct the desired circuit, we build separate circuits for $p \vee \neg r$ and for $\neg p \vee (q \vee \neg r)$ and combine them using an AND gate.

To construct a circuit for $p \vee \neg r$, we use an inverter to produce $\neg r$ from the input r. Then, we use an OR gate to combine p and $\neg r$.

To build a circuit for $\neg p \vee (q \vee \neg r)$, we first use an inverter to obtain $\neg r$.

Then we use an OR gate with inputs q and $\neg r$ to obtain $q \vee \neg r$.

Finally, we use another inverter and an OR gate to get $\neg p \vee (q \vee \neg r)$ from the inputs p and $q \vee \neg r$.

To complete the construction, we employ a final AND gate, with inputs $p \vee \neg r$ and $\neg p \vee (q \vee \neg r)$.

SOLUTION

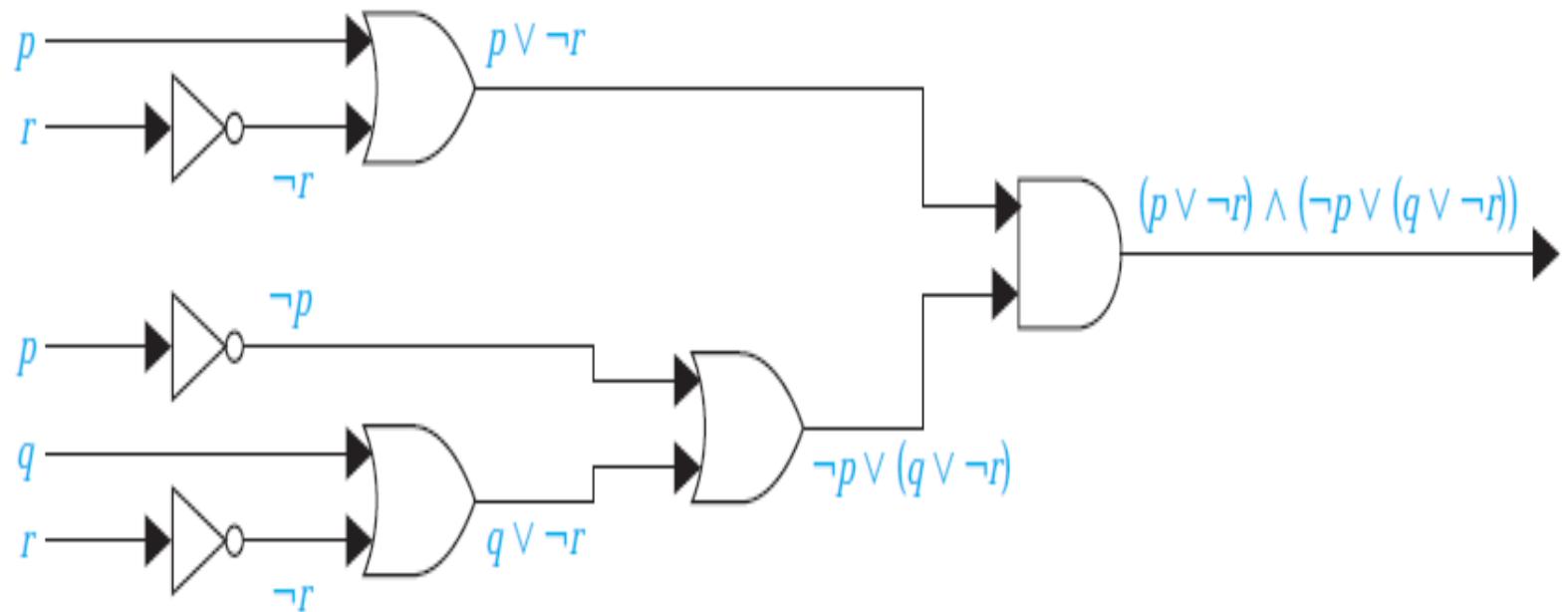
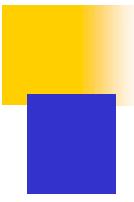
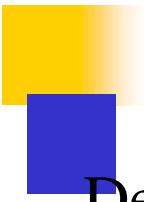


FIGURE 3 The circuit for $(p \vee \neg r) \wedge (\neg p \vee (q \vee \neg r))$.



Propositional Satisfiability

- A compound proposition is **satisfiable** if there is an assignment of truth values to its variables that makes it true.
- compound proposition is **unsatisfiable** if and only if its negation is true for all assignments of truth values to the variables, that is, if and only if its negation is a **tautology**.



EXAMPLE

Determine whether each of the compound propositions

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

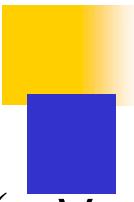
$$(p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r), \text{ and}$$

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

is satisfiable.

SOL:

- $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p)$ is true when the three variable p, q, and r have the same truth value.
- Hence, it is satisfiable as there is at least one assignment of truth values for p, q, and r that makes it true.
- $(p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$ is true when at least one of p, q, and r is true and at least one is false.
- Hence, $(p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$ is satisfiable, as there is at least one assignment of truth values for p, q, and r that makes it true.



EXAMPLE con...

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

to be true, $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p)$ and $(p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$ must both be true.

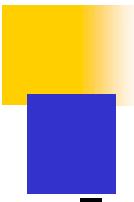
For the first to be true, the three variables must have the same truth values,

for the second to be true, at least one of three variables must be true and at least one must be false.

these conditions are contradictory. From these observations we conclude that no assignment of truth values to p , q , and r makes

$$(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p) \wedge (p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$$
 true.

Hence, it is unsatisfiable.

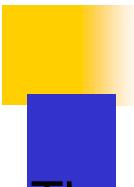


Predicates

- Propositional logic, cannot adequately express the meaning of all statements in mathematics and in natural language.
- For example, suppose that we know that
“Every computer connected to the university network is functioning properly.”

Statements involving variables, such as
“ $x > 3$,” “ $x = y + 3$,” “ $x + y = z$,”
and
“computer x is *under attack by an intruder*,”
and
“computer x is *functioning properly*,”

These statements are neither true nor false when the values of the variables are not specified.
propositions can be produced from such statements.



Predicates

The statement “*x is greater than 3*” has two parts.

The first part, the variable x, is the subject of the statement.

The second part—the predicate, “is greater than 3”—refers to a property that the subject of the statement can have.

We can denote the statement “*x is greater than 3*” by $P(x)$, where P denotes the predicate “is greater than 3” and x is the variable. The statement $P(x)$ is also said to be the value of the propositional function P at x .

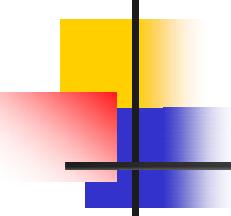
Once a value has been assigned to the variable x , the statement $P(x)$ becomes a proposition and has a truth value.

EXAMPLE

Let $P(x)$ denote the statement “ $x > 3$.” What are the truth values of $P(4)$ and $P(2)$?

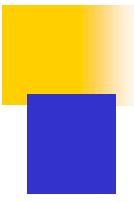
EXAMPLE

Let $A(x)$ denote the statement “Computer x is under attack by an intruder.” Suppose that of the computers on campus, only CS2 and MATH1 are currently under attack by intruders. What are truth values of $A(\text{CS1})$, $A(\text{CS2})$, and $A(\text{MATH1})$?



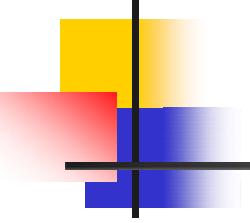
Previously...

- In predicate logic, a ***predicate*** is modeled as a ***propositional function* $P(\cdot)$** from subjects to propositions.
 - $P(x)$: “ x is a prime number” (x : any subject)
 - $P(3)$: “3 is a prime number.” (proposition!)
- Propositional functions of **any** number of arguments, each of which may take **any** grammatical role that a noun can take
 - $P(x,y,z)$: “ **x** gave **y** the grade **z** ”
 - $P(\text{Mike}, \text{Mary}, \text{A})$: “**Mike** gave **Mary** the grade **A**.”



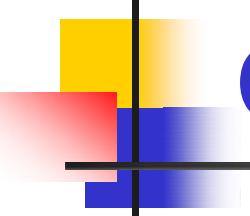
Quantifiers

- Quantification expresses the extent to which a predicate is true over a range of elements. In English, the words all, some, many, none, and few are used in quantifications.
- universal quantification- predicate is true for every element under consideration.
- existential quantification- there is one or more element under consideration for which the predicate is true.



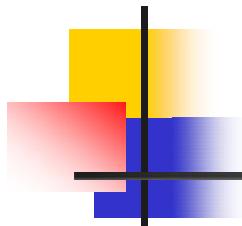
Universe of Discourse (U.D.)

- The power of distinguishing subjects from predicates is that it lets you state things about *many* objects at once.
- e.g., let $P(x) = "x + 1 > x"$. We can then say, “For **any** number x , $P(x)$ is true” instead of $(0 + 1 > 0) \wedge (1 + 1 > 1) \wedge (2 + 1 > 2) \wedge \dots$
- The collection of values that a variable x can take is called x ’s ***universe of discourse*** or the ***domain of discourse***.



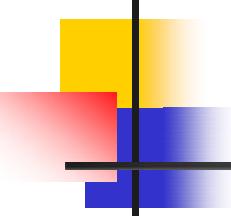
Quantifier Expressions

- **Quantifiers** provide a notation that allows us to *quantify* (count) *how many* objects in the universe of discourse satisfy the given predicate.
- “ \forall ” is the FOR ALL or *universal* quantifier.
 $\forall x P(x)$ means for all x in the domain, $P(x)$.
- “ \exists ” is the EXISTS or *existential* quantifier.
 $\exists x P(x)$ means there exists an x in the domain
(that is, 1 or more) such that $P(x)$.



The Universal Quantifier \forall

- $\forall x P(x)$: *For all x in the domain, $P(x)$.*
 - $\forall x P(x)$ is
 - *true* if $P(x)$ is true for every x in D (D : domain of discourse)
 - *false* if $P(x)$ is false for at least one x in D
 - For every real number x , $x^2 \geq 0$ TRUE
 - For every real number x , $x^2 - 1 > 0$ FALSE
 - A **counterexample** to the statement $\forall x P(x)$ is a value x in the domain D that makes $P(x)$ false
 - What is the truth value of $\forall x P(x)$ when the domain is empty? TRUE

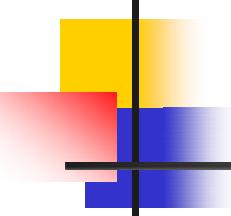


The Universal Quantifier \forall

- If all the elements in the domain can be listed as x_1, x_2, \dots, x_n then, $\forall x P(x)$ is the same as the conjunction:

$$P(x_1) \wedge P(x_2) \wedge \cdots \wedge P(x_n)$$

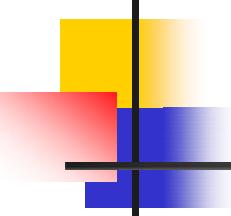
- Example: Let the domain of x be parking spaces at UH. Let $P(x)$ be the statement “ x is full.” Then the ***universal quantification*** of $P(x)$, $\forall x P(x)$, is the *proposition*:
 - “All parking spaces at UH are full.”
 - or “Every parking space at UH is full.”
 - or “For each parking space at UH, that space is full.”



The Existential Quantifier \exists

- $\exists x P(x)$: *There exists an x in the domain (that is, 1 or more) such that $P(x)$.*
- $\exists x P(x)$ is
 - *true* if $P(x)$ is true for at least one x in the domain
 - *false* if $P(x)$ is false for every x in the domain
- What is the truth value of $\exists x P(x)$ when the domain is empty? FALSE
- If all the elements in the domain can be listed as x_1, x_2, \dots, x_n then, $\exists x P(x)$ is the same as the disjunction:

$$P(x_1) \vee P(x_2) \vee \dots \vee P(x_n)$$



The Existential Quantifier \exists

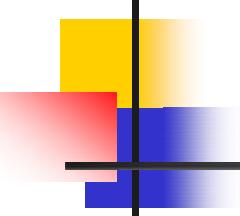
- Example:

Let the domain of x be parking spaces at UH.

Let $P(x)$ be the statement “ x is full.”

Then the ***existential quantification*** of $P(x)$,
 $\exists x P(x)$, is the *proposition*:

- “Some parking spaces at UH are full.”
- or “There is a parking space at UH that is full.”
- or “At least one parking space at UH is full.”



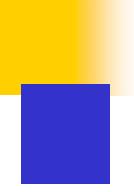
Free and Bound Variables

- An expression like $P(x)$ is said to have a ***free variable*** x (meaning, x is undefined).
- A quantifier (either \forall or \exists) *operates* on an expression having one or more free variables, and ***binds*** one or more of those variables, to produce an expression having one or more ***bound variables***.

Example of Binding

- $P(x,y)$ has 2 free variables, x and y .
- $\forall x P(x,y)$ has 1 free variable  , and one bound variable  . [Which is which?]
 - “ $P(x)$, where $x = 3$ ” is another way to bind x .
 - An expression with zero free variables is a bona-fide (actual) proposition.
 - An expression with one or more free variables is not a proposition:

e.g. $\forall x P(x,y) = Q(y)$



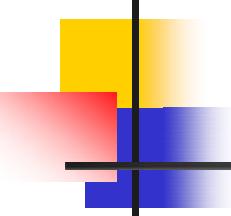
Scope

- The statement $\exists x(x + y = 1)$, *the variable x is bound by the existential quantification $\exists x$, but the variable y is free because it is not bound by a quantifier and no value is assigned to this variable.* This illustrates that in the statement $\exists x(x + y = 1)$, *x is bound, but y is free.*

Scope

$\exists x(P(x) \wedge Q(x)) \vee \forall xR(x)$ - all variables are bound

The scope of the first quantifier, $\exists x$, *is the expression $P(x) \wedge Q(x)$ because $\exists x$ is applied only to $P(x) \wedge Q(x)$, and not to the rest of the statement.* Similarly, the scope of the second quantifier, $\forall x$, *is the expression $R(x)$.*



Quantifiers with Restricted Domains

- An abbreviated notation is often used to restrict the domain of a quantifier, a condition a variable must satisfy is included after the quantifier.
- $\forall x > 0 P(x)$ is shorthand for

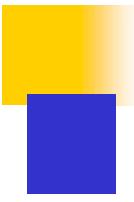
“For all x that are greater than zero, $P(x)$.”

$$= \forall x (x > 0 \rightarrow P(x))$$

- $\exists x > 0 P(x)$ is shorthand for

“There is an x greater than zero such that $P(x)$.”

$$= \exists x (x > 0 \wedge P(x))$$



Example

What do the statements $\forall x < 0 (x^2 > 0)$, $\forall y = 0 (y^3 = 0)$, and $\exists z > 0 (z^2 = 2)$ mean, where the domain in each case consists of the real numbers?

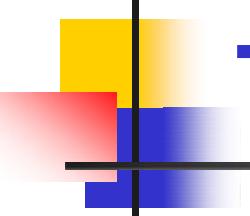
Solution: The statement $\forall x < 0 (x^2 > 0)$ states that for every real number x with $x < 0$, $x^2 > 0$.

That is, it states “The square of a negative real number is positive.” This statement is the same as $\forall x(x < 0 \rightarrow x^2 > 0)$.

The statement $\forall y = 0 (y^3 = 0)$ states that for every real number y with $y = 0$, we have $y^3 = 0$. That is, it states “The cube of every nonzero real number is nonzero.”

Note that this statement is equivalent to $\forall y(y = 0 \rightarrow y^3 = 0)$.

Finally, the statement $\exists z > 0 (z^2 = 2)$ states that there exists a real number z with $z > 0$ such that $z^2 = 2$. That is, it states “There is a positive square root of 2.” This statement is equivalent to $\exists z(z > 0 \wedge z^2 = 2)$.

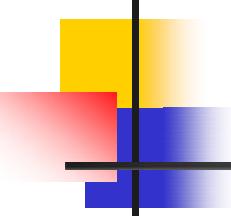


Translating from English

- Express the statement “*Every student in this class has studied calculus*” using predicates and quantifiers.
 - Let $C(x)$ be the statement: “*x has studied calculus.*”
 - If domain for x consists of the students in this class, then
 - it can be translated as $\forall x C(x)$

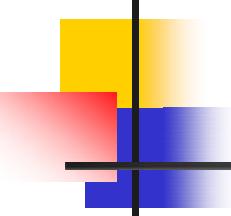
or

- If domain for x consists of all people
- Let $S(x)$ be the predicate: “*x is in this class*”
- Translation: $\forall x (S(x) \rightarrow C(x))$



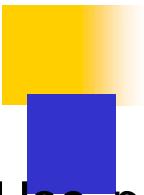
Translating from English

- Express the statement “*Some students in this class has visited Mexico*” using predicates and quantifiers.
 - Let $M(x)$ be the statement: “*x has visited Mexico*”
 - If domain for x consists of the students in this class, then
 - it can be translated as $\exists x M(x)$ or
 - If domain for x consists of all people
 - Let $S(x)$ be the statement: “*x is in this class*”
 - Then, the translation is $\exists x (S(x) \wedge M(x))$



Translating from English

- Express the statement “*Every student in this class has visited either Canada or Mexico*” using predicates and quantifiers.
 - Let $C(x)$ be the statement: “ x has visited Canada” and $M(x)$ be the statement: “ x has visited Mexico”
 - If domain for x consists of the students in this class, then
 - it can be translated as $\forall x (C(x) \vee M(x))$



Example

Q. Use predicates and quantifiers to express the system specifications “Every mail message larger than one megabyte will be compressed” and “If a user is active, at least one network link will be available.”

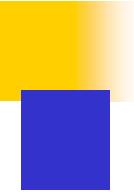
Solution: Let $S(m, y)$ be “Mail message m is larger than y megabytes,” where the variable x has the domain of all mail messages and the variable y is a positive real number, and let $C(m)$ denote “Mail message m will be compressed.” Then the specification “Every mail message larger than one megabyte will be compressed” can be represented as $\forall m(S(m, 1) \rightarrow C(m))$.

Let $A(u)$ represent “User u is active,” where the variable u has the domain of all users,

let $S(n, x)$ denote “Network link n is in state x ,” where n has the domain of all network links and x has the domain of all possible states for a network link.

Then the specification

“If a user is active, at least one network link will be available” can be represented by $\exists u A(u) \rightarrow \exists n S(n, \text{available})$.



Example

Q. Consider these statements. The first two are called *premises and the third is called the conclusion.*

The entire set is called an *argument*.

“All lions are fierce.”

“Some lions do not drink coffee.”

“Some fierce creatures do not drink coffee.”

Solution: We can express these statements as:

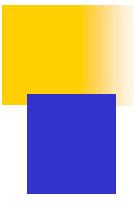
$$\forall x(P(x) \rightarrow Q(x)).$$

$$\exists x(P(x) \wedge \neg R(x)).$$

$$\exists x(Q(x) \wedge \neg R(x)).$$

Notice that the second statement cannot be written as $\exists x(P(x) \rightarrow \neg R(x))$.

The reason is that $P(x) \rightarrow \neg R(x)$ is true whenever x is not a lion, so that $\exists x(P(x) \rightarrow \neg R(x))$ is true as long as there is at least one creature that is not a lion, even if every lion drinks coffee. Similarly, the third statement cannot be written as $\exists x(Q(x) \rightarrow \neg R(x))$.



Example

Consider these statements, of which the first three are premises and the fourth is a valid conclusion.

“All hummingbirds are richly colored.”

“No large birds live on honey.”

“Birds that do not live on honey are dull in color.”

“Hummingbirds are small.”

Let $P(x)$, $Q(x)$, $R(x)$, and $S(x)$ be the statements “ x is a hummingbird,” “ x is large,” “ x lives on honey,” and “ x is richly colored,” respectively. Assuming that the domain consists of all birds, express the statements in the argument using quantifiers and $P(x)$, $Q(x)$, $R(x)$, and $S(x)$.

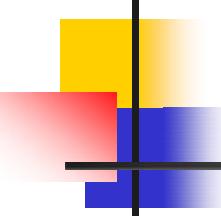
Solution: We can express the statements in the argument as

$$\forall x(P(x) \rightarrow S(x)).$$

$$\neg \exists x(Q(x) \wedge R(x)).$$

$$\forall x(\neg R(x) \rightarrow \neg S(x)).$$

$$\forall x(P(x) \rightarrow \neg Q(x)).$$



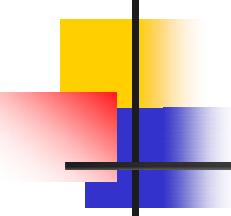
Negations of Quantifiers

- $\forall x P(x)$: “Every student in the class has taken a course in calculus” ($P(x)$: “ x has taken a course in calculus”)
 - “Not every student in the class ... calculus”

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

- Consider $\exists x P(x)$: “There is a student in the class who has taken a course in calculus”
 - “There is no student in the class who has taken a course in calculus”

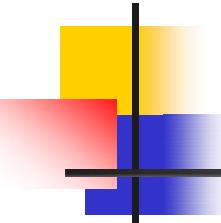
$$\neg \exists x P(x) \equiv \forall x \neg P(x)$$



Negations of Quantifiers

- Definitions of quantifiers: If the domain = $\{a, b, c, \dots\}$
 - $\forall x P(x) \equiv P(a) \wedge P(b) \wedge P(c) \wedge \dots$
 - $\exists x P(x) \equiv P(a) \vee P(b) \vee P(c) \vee \dots$
- From those, we can prove the laws:
 - $\neg \forall x P(x) \equiv \neg(P(a) \wedge P(b) \wedge P(c) \wedge \dots)$
 $\equiv \neg P(a) \vee \neg P(b) \vee \neg P(c) \vee \dots$
 $\equiv \exists x \neg P(x)$
 - $\neg \exists x P(x) \equiv \neg(P(a) \vee P(b) \vee P(c) \vee \dots)$
 $\equiv \neg P(a) \wedge \neg P(b) \wedge \neg P(c) \wedge \dots$
 $\equiv \forall x \neg P(x)$
- Which *propositional* equivalence law was used to prove this?

DeMorgan's
Demorgan's



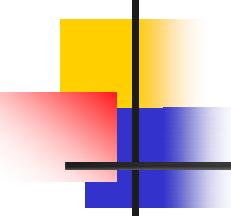
Negations of Quantifiers

Theorem:

■ Generalized De Morgan's laws for logic

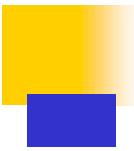
$$1. \neg \forall x P(x) \equiv \exists x \neg P(x)$$

$$2. \neg \exists x P(x) \equiv \forall x \neg P(x)$$



Negations: Examples

- What are the negations of the statements $\forall x (x^2 > x)$ and $\exists x (x^2 = 2)$?
 - $\neg \forall x (x^2 > x) \equiv \exists x \neg(x^2 > x) \equiv \exists x (x^2 \leq x)$
 - $\neg \exists x (x^2 = 2) \equiv \forall x \neg(x^2 = 2) \equiv \forall x (x^2 \neq 2)$
- Show that $\neg \forall x(P(x) \rightarrow Q(x))$ and $\exists x(P(x) \wedge \neg Q(x))$ are logically equivalent.
 - $\neg \forall x(P(x) \rightarrow Q(x)) \equiv \exists x \neg(P(x) \rightarrow Q(x))$
 $\equiv \exists x \neg(\neg P(x) \vee Q(x))$
 - $\equiv \exists x (P(x) \wedge \neg Q(x))$



Summary

© The McGraw-Hill Companies, Inc. all rights reserved.

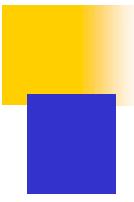
TABLE 1 Quantifiers.

<i>Statement</i>	<i>When True?</i>	<i>When False?</i>
$\forall x P(x)$	$P(x)$ 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 .

© The McGraw-Hill Companies, Inc. all rights reserved.

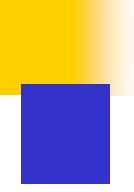
TABLE 2 De Morgan's Laws for Quantifiers.

<i>Negation</i>	<i>Equivalent Statement</i>	<i>When Is Negation True?</i>	<i>When False?</i>
$\neg \exists x P(x)$	$\forall x \neg P(x)$	For every x , $P(x)$ is false.	There is an x for which $P(x)$ is true.
$\neg \forall x P(x)$	$\exists x \neg P(x)$	There is an x for which $P(x)$ is false.	$P(x)$ is true for every x .



Logic Programming

- A programming language is designed to reason using the rules of predicate logic. Prolog (from *Programming in Logic*)
- Prolog programs include a set of declarations consisting of two types of statements.
- **Prolog facts and Prolog rules. Prolog facts define predicates by specifying the elements that satisfy these predicates.**
- Prolog rules are used to define new predicates using those already defined by Prolog facts.



Example

Consider a Prolog program given facts telling it the instructor of each class and in which classes students are enrolled. The program uses these facts to answer queries concerning the professors who teach particular students. Such a program could use the predicates *instructor(p, c)* and *enrolled(s, c)* to represent that professor *p* is the instructor of course *c* and that student *s* is enrolled in course *c*, respectively. For example, the Prolog facts in such a program might include:

instructor(chan,math273)

instructor(patel,ee222)

instructor(grossman,cs301)

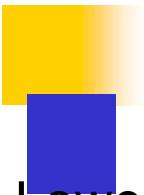
enrolled(kevin,math273)

enrolled(juana,ee222)

enrolled(juana,cs301)

enrolled(kiko,math273)

enrolled(kiko,cs301)



Example

Lowercase letters have been used for entries because Prolog considers names beginning with an uppercase letter to be variables.

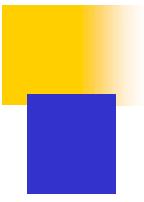
A new predicate *teaches(p, s)*, representing that professor *p* teaches student *s*, can be defined using the Prolog rule

teaches(P,S) :- instructor(P,C), enrolled(S,C)

teaches(p, s) is true if there exists a class *c* such that professor *p* is the instructor of class *c* and student *s* is enrolled in class *c*.

(Note that a comma is used to represent a conjunction of predicates in Prolog. Similarly, a semicolon is used to represent a disjunction of predicates.)

Prolog answers queries using the facts and rules it is given. For example, using the facts and rules listed, the query ?enrolled(kevin,math273) produces the response yes



Example

?enrolled(X,math273)

kevin

Kiko

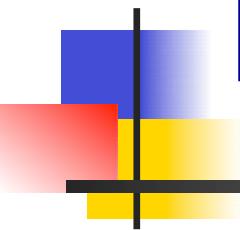
To find all the professors who are instructors in classes being taken by Juana, we use the query

?teaches(X,juana)

This query returns

patel

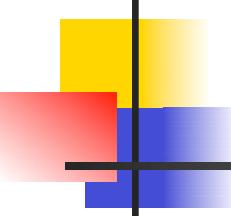
grossman



Lecture 5

Chapter 1. The Foundations

- 4. Nested Quantifiers
- 5. Rules of Inference



Nesting of Quantifiers

■■ Example:

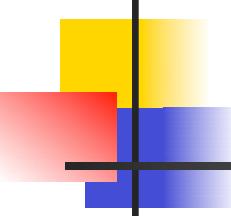
Let the domain of x and y be people.

Let $L(x,y)$ = “ x likes y ” (A statement with 2 free variables – not a proposition)

■■ Then $\exists y L(x,y)$ = “There is someone whom x likes.” (A statement with 1 free variable x – not a proposition)

■■ Then $\forall x (\exists y L(x,y))$ =

“Everyone has someone whom they like.” (A _____ with _____ *Proposition* variables.) Ø



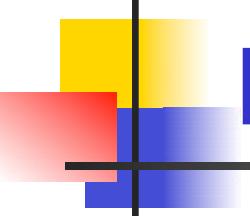
Nested Quantifiers

- Nested quantifiers are quantifiers that occur within the scope of other quantifiers.
- The order of the quantifiers is important, unless all the quantifiers are universal quantifiers or all are existential quantifiers.

© The McGraw-Hill Companies, Inc. all rights reserved.

TABLE 1 Quantifications of Two Variables.

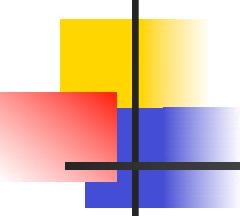
<i>Statement</i>	<i>When True?</i>	<i>When False?</i>
$\forall x \forall y P(x, y)$ $\forall y \forall x P(x, y)$	$P(x, y)$ is true for every pair x, y .	There is a pair x, y for which $P(x, y)$ is false.
$\forall x \exists y P(x, y)$	For every x there is a y for which $P(x, y)$ is true.	There is an x such that $P(x, y)$ 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.
$\exists x \exists y P(x, y)$ $\exists y \exists x P(x, y)$	There is a pair x, y for which $P(x, y)$ is true.	$P(x, y)$ is false for every pair x, y .



Nested Quantifiers

- Let the domain of x and y is \mathbf{R} , and $P(x,y)$: $xy = 0$.
Find the truth value of the following propositions.
 - $\forall x \forall y P(x, y)$ (F)
 - $\forall x \exists y P(x, y)$ (T)
 - $\exists x \forall y P(x, y)$ (T)
 - $\exists x \exists y P(x, y)$ (T)
- $\forall x \exists y P(x,y) \not\equiv \exists y \forall x P(x,y)$
 - For every x , there exists y such that $x + y = 0$. (T)
 - There exists y such that, for every x , $x + y = 0$. (F)

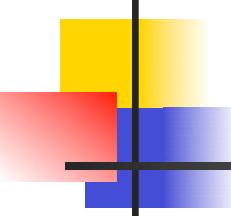
\mathbf{R} : set of real
numbers



Nested Quantifiers: Example

- Let the domain = {1, 2, 3}. Find an expression equivalent to $\forall x \exists y P(x,y)$ where the variables are bound by substitution instead:
 - Expand from inside out or outside in.
 - Outside in:

$$\begin{aligned}\forall x \exists y P(x,y) \\ \equiv \exists y P(1,y) \wedge \exists y P(2,y) \wedge \exists y P(3,y) \\ \equiv [P(1,1) \vee P(1,2) \vee P(1,3)] \wedge \\ [P(2,1) \vee P(2,2) \vee P(2,3)] \wedge \\ [P(3,1) \vee P(3,2) \vee P(3,3)]\end{aligned}$$



Quantifier Exercise

- If $R(x,y)$ = “ x relies upon y ,” express the following in unambiguous English when the domain is all people

$$\forall x(\exists y R(x,y)) = \text{Everyone has } \textit{someone} \text{ to rely on.}$$

$$\exists y(\forall x R(x,y)) =$$

There's a poor overburdened soul whom *everyone* relies upon (including himself)!

$$\exists x(\forall y R(x,y)) =$$

There's some needy person who relies upon *everybody* (including himself).

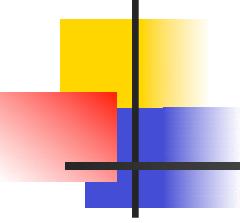
$$\forall y(\exists x R(x,y)) =$$

Everyone has *someone* who relies upon them.

$$\forall x(\forall y R(x,y)) =$$

Everyone relies upon *everybody*, (including themselves)!



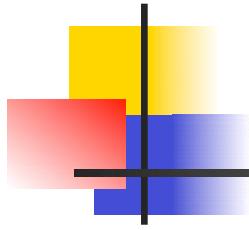


Negating Nested Quantifiers

- Successively apply the rules for negating statements involving a single quantifier

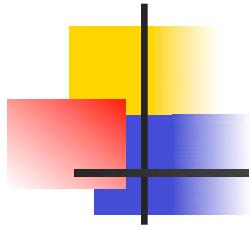
Example: Express the negation of the statement $\forall x \exists y (P(x,y) \wedge \exists z R(x,y,z))$ so that all negation symbols immediately precede predicates.

$$\begin{aligned}& \neg \forall x \exists y (P(x,y) \wedge \exists z R(x,y,z)) \\& \equiv \exists x \neg \exists y (P(x,y) \wedge \exists z R(x,y,z)) \\& \equiv \exists x \forall y \neg (P(x,y) \wedge \exists z R(x,y,z)) \\& \equiv \exists x \forall y (\neg P(x,y) \vee \neg \exists z R(x,y,z)) \\& \equiv \exists x \forall y (\neg P(x,y) \vee \forall z \neg R(x,y,z))\end{aligned}$$



Equivalence Laws

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



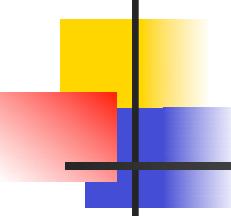
Notational Conventions

- Quantifiers have higher precedence than all logical operators from propositional logic:

$$(\forall x P(x)) \wedge Q(x)$$

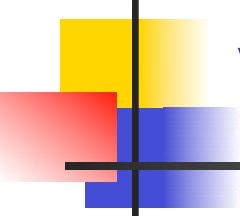
- Consecutive quantifiers of the same type can be combined:

$$\forall x \forall y \forall z P(x,y,z) \equiv \forall x, y, z P(x,y,z) \text{ or even}$$
$$\forall xyz P(x,y,z)$$



1.5 Rules of Inference

- **An argument:** a sequence of statements that end with a conclusion
- Some forms of argument (“valid”) never lead from correct statements to an incorrect conclusion. Some other forms of argument (“fallacies”) can lead from true statements to an incorrect conclusion.
- **A logical argument** consists of a list of (possibly compound) propositions called premises/hypotheses and a single proposition called the conclusion.
- **Logical rules of inference:** methods that depend on logic alone for deriving a new statement from a set of other statements. (Templates for constructing valid arguments.)



Valid Arguments (I)

- Example: A logical argument

If I dance all night, then I get tired. I danced all night.

Therefore I got tired.

- Logical representation of underlying variables:

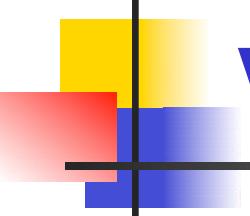
p : I dance all night. q : I get tired.

- Logical analysis of argument:

$p \rightarrow q$ premise 1

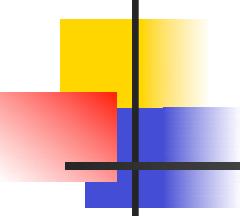
p premise 2

$\therefore q$ conclusion



Valid Arguments (II)

- A form of logical argument is ***valid*** if whenever every premise is true, the conclusion is also true. A form of argument that is not valid is called a ***fallacy***.



Inference Rules: General Form

■■ An *Inference Rule* is

■■ A pattern establishing that if we know that a set of *premise* statements of certain forms are all true, then we can validly deduce that a certain related *conclusion* statement is true.

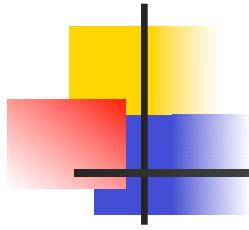
premise 1

premise 2

...

∴ conclusion

“∴” means “therefore”



Inference Rules & Implications

- Each valid logical inference rule corresponds to an implication that is a tautology.

Inference rule

<i>premise 1</i>
<i>premise 2</i>
...
<hr/> <i>∴ conclusion</i>

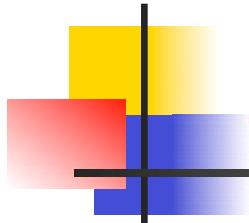
- Corresponding tautology:
 $((\text{premise 1}) \wedge (\text{premise 2}) \wedge \dots) \rightarrow \text{conclusion}$

Modus Ponens

- $$\begin{array}{c} p \\ p \rightarrow q \\ \hline \therefore q \end{array}$$
- Rule of **Modus ponens**
(a.k.a. *law of detachment*)
- “the mode of affirming”
- $(p \wedge (p \rightarrow q)) \rightarrow q$ is a tautology

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

- Notice that the first row is the only one where premises are all true



Modus Ponens: Example

If $\left\{ \begin{array}{l} p \rightarrow q : \text{"If it snows today} \\ \quad \text{then we will go skiing"} \\ p : \text{"It is snowing today"} \end{array} \right\}$ assumed TRUE
Then $\frac{}{\therefore q} : \text{"We will go skiing"}$ is TRUE

If $\left\{ \begin{array}{l} p \rightarrow q : \text{"If } n \text{ is divisible by 3} \\ \quad \text{then } n^2 \text{ is divisible by 3"} \\ p : \text{"} n \text{ is divisible by 3"} \end{array} \right\}$ assumed TRUE
Then $\frac{}{\therefore q} : \text{"} n^2 \text{ is divisible by 3"}$ is TRUE

Modus Tollens

- $$\begin{array}{c} \neg q \\ \hline p \rightarrow q \\ \therefore \neg p \end{array}$$

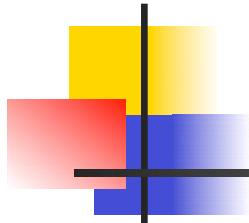
Rule of ***Modus tollens***

“the mode of denying”

■■ $(\neg q \wedge (p \rightarrow q)) \rightarrow \neg p$ is a tautology

■■ Example

If $\left. \begin{array}{l} p \rightarrow q : \text{“If this jewel is really a diamond} \\ \text{then it will scratch glass”} \\ \\ \neg q \quad : \text{“The jewel doesn’t scratch glass”} \end{array} \right\} \text{assumed TRUE}$
Then $\frac{}{\therefore \neg p} : \text{“The jewel is not a diamond”}$ is TRUE



More Inference Rules

...	p
	$\therefore p \vee q$

Rule of **Addition**

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

...	$p \wedge q$
	$\therefore p$

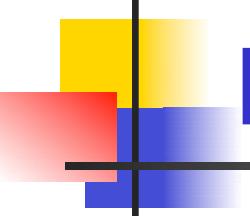
Rule of **Simplification**

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

...	p
	q
	$\therefore p \wedge q$

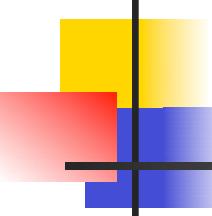
Rule of **Conjunction**

Tautology: $[(p) \wedge (q)] \rightarrow p \wedge q$



Examples

- State which rule of inference is the basis of the following arguments:
 - It is below freezing now. Therefore, it is either below freezing or raining now.
 - It is below freezing and raining now. Therefore, it is below freezing now.
- p : It is below freezing now.
 q : It is raining now.
 - $p \rightarrow (p \vee q)$ (rule of addition)
 - $(p \wedge q) \rightarrow p$ (rule of simplification)



Hypothetical Syllogism

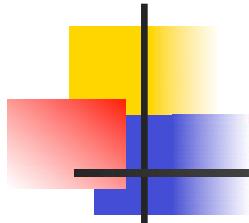
- $$\begin{array}{c} p \rightarrow q \\ q \rightarrow r \\ \hline \therefore p \rightarrow r \end{array}$$

Rule of *Hypothetical syllogism*
Tautology:

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

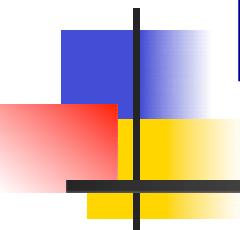
- Example: State the rule of inference used in the argument:

“If it rains today, then we will not have a barbecue today. If we do not have a barbecue today, then we will have a barbecue tomorrow. Therefore, if it rains today, then we will have barbecue tomorrow.”



Disjunctive Syllogism

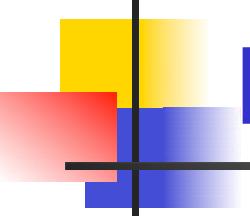
- $$\begin{array}{c} p \vee q \\ \neg p \\ \hline \therefore q \end{array}$$
 Rule of ***Disjunctive syllogism***
- Example
 - Ed's wallet is in his back pocket or it is on his desk. ($p \vee q$) p q
 - Ed's wallet is not in his back pocket. ($\neg p$)
 - Therefore, Ed's wallet is on his desk. (q)



Lecture 6

Chapter 1. The Foundations

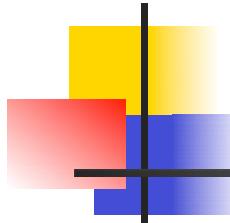
1.5 Rules of Inference



Previously...

- Rules of inference
 - Modus ponens
 - Modus tollens
 - Hypothetical syllogism
 - Disjunctive syllogism
 - Resolution
 - Addition
 - Simplification
 - Conjunction

Table 1 in pp.66



Resolution

$$\begin{array}{c} p \vee q \\ \neg p \vee r \\ \hline \therefore q \vee r \end{array}$$

Rule of ***Resolution***

Tautology:

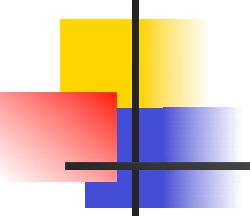
$$[(p \vee q) \wedge (\neg p \vee r)] \rightarrow (q \vee r)$$

■ When $q = r$:

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

■ When $r = \text{F}$:

$$[(p \vee q) \wedge (\neg p)] \rightarrow q \quad (\text{Disjunctive syllogism})$$



Resolution: Example

$$\begin{array}{c} p \vee q \\ \neg p \vee r \\ \hline \therefore q \vee r \end{array}$$

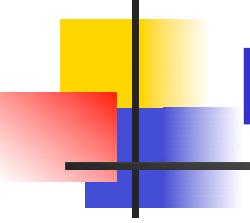
■■ Example: Use resolution to show that the hypotheses “Jasmine is skiing or it is not snowing” and “It is snowing or Bart is playing hockey” imply that “Jasmine is skiing or Bart is playing hockey”

$p \vee q$ $\neg p \vee r$ $q \vee r$

r $\neg p$ p

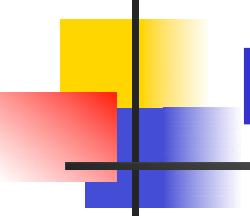
$\neg p$ r q

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



Formal Proofs

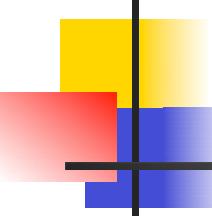
- A formal proof of a conclusion C , given premises p_1, p_2, \dots, p_n consists of a sequence of *steps*, each of which applies some inference rule to premises or previously-proven statements to yield a new true statement (the *conclusion*).
- A proof demonstrates that *if* the premises are true, *then* the conclusion is true.



Formal Proof Example

- Suppose we have the following premises:
“It is not sunny and it is cold.” “We will swim only if it is sunny.”
“If we do not swim, then we will canoe.”
“If we canoe, then we will be home by sunset.”

- Given these premises, prove the conclusion
“We will be home by sunset” using inference rules.



Proof Example cont.

- Step 1: Identify the propositions (Let us adopt the following abbreviations)
 - $sunny$ = “**It is sunny**”; $cold$ = “**It is cold**”; $swim$ = “**We will swim**”; $canoe$ = “**We will canoe**”; $sunset$ = “**We will be home by sunset**”.
- Step 2: Identify the argument. (Build the argument form)
 - $\neg sunny \wedge cold$
 - $swim \rightarrow sunny$
 - $\neg swim \rightarrow canoe$
 - $canoe \rightarrow sunset$

$\therefore sunset$

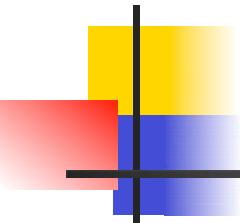
It is not sunny and it is cold.

We will swim only if it is sunny.

If we do not swim, then we will canoe.

If we canoe, then we will be home by sunset.

We will be home by sunset.



Proof Example *cont.*

- Step 3: Verify the reasoning using the rules of inference

Step

1. $\neg \text{sunny} \wedge \text{cold}$
2. $\neg \text{sunny}$
3. $\text{swim} \rightarrow \text{sunny}$
4. $\neg \text{swim}$
5. $\neg \text{swim} \rightarrow \text{canoe}$
6. canoe
7. $\text{canoe} \rightarrow \text{sunset}$
8. sunset

Proved by

Premise #1.

Simplification of 1.

Premise #2.

Modus tollens on 2 and 3.

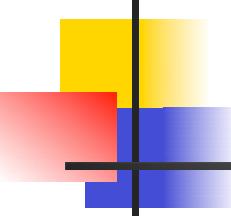
Premise #3.

Modus ponens on 4 and 5.

Premise #4.

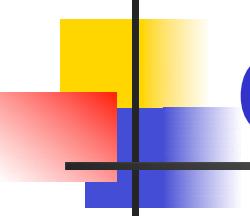
Modus ponens on 6 and 7.

$$\begin{array}{c} \neg \text{sunny} \wedge \text{cold} \\ \text{swim} \rightarrow \text{sunny} \\ \neg \text{swim} \rightarrow \text{canoe} \\ \hline \text{canoe} \rightarrow \\ \hline \text{sunset} \\ \therefore \text{sunset} \end{array}$$



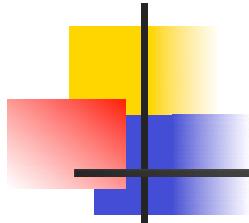
Common Fallacies

- A ***fallacy*** is an inference rule or other proof method that is not logically valid.
- A fallacy may yield a false conclusion!
- ***Fallacy of affirming the conclusion:***
 - “ $p \rightarrow q$ is true, and q is true, so p must be true.” (No, because $\text{F} \rightarrow \text{T}$ is true.)
- Example
 - If David Cameron (DC) is president of the US, then he is at least 40 years old. ($p \rightarrow q$)
 - DC is at least 40 years old. (q)
 - Therefore, DC is president of the US. (p)



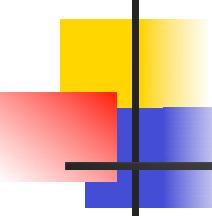
Common Fallacies (*cont'd*)

- *Fallacy of denying the hypothesis:*
- “ $p \rightarrow q$ is true, and p is false, so q must be false.” (No, again because $\text{F} \rightarrow \text{T}$ is true.)
- Example
 - If a person does arithmetic well then his/her checkbook will balance. ($p \rightarrow q$)
 - I cannot do arithmetic well. ($\neg p$)
 - Therefore my checkbook does not balance. ($\neg q$)



Inference Rules for Quantifiers

- $$\frac{\forall x P(x)}{\therefore P(c)}$$
 (substitute any specific member c in the domain) **Universal instantiation**
- $$\frac{P(\theta)}{\therefore \forall x P(x)}$$
 (for an arbitrary element c of the domain) **Universal generalization**
- $$\frac{\exists x P(x)}{\therefore P(c)}$$
 (substitute an element c for which $P(c)$ is true) **Existential instantiation**
- $$\frac{P(\theta)}{\therefore \exists x P(x)}$$
 (for some element c in the domain) **Existential generalization**



Example

■■ Every man has two legs. John Smith is a man.
Therefore, John Smith has two legs.

■■ Proof

■■ Define the predicates:

■■ $M(x)$: x is a man

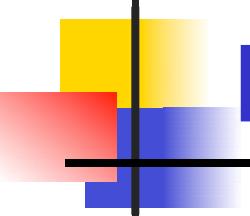
■■ $L(x)$: x has two legs

■■ J : John Smith, a member of the universe

■■ The argument becomes 1. $\forall x [M(x) \rightarrow L(x)]$

2. $M(J)$

$\therefore L(J)$



Example cont.

$$\forall x (M(x) \rightarrow L(x))$$

$$M(J)$$

$$\therefore L(J)$$

■ The proof is

1. $\forall x [M(x) \rightarrow L(x)]$

Premise 1

2. $M(J) \rightarrow L(J)$

U. I. from (1)

3. $M(J)$

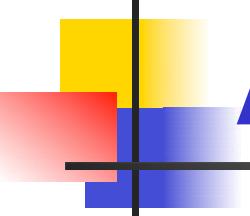
Premise 2

4. $L(J)$

Modus Ponens from (2) and (3)

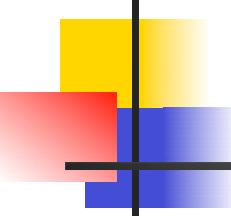
■ Note: Using the rules of inference requires lots of practice.

■ Try example problems in the textbook.



Another example

- Correct or incorrect: “At least one of the 20 students in the class is intelligent. John is a student of this class. Therefore, John is intelligent.”
- First: Separate premises from conclusion
- *Premises:*
 1. At least one of the 20 students in the class is intelligent.
 2. John is a student of this class.
- *Conclusion:* John is intelligent.



Answer

- Next, translate the example in logic notation.
- Premise 1: At least one of the 20 students in the class is intelligent.

Let the domain = all people

$C(x)$ = “ x is in the class”

$I(x)$ = “ x is intelligent”

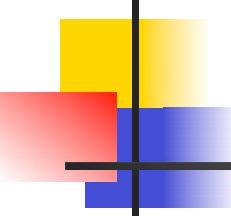
Then *Premise 1* says: $\exists x(C(x) \wedge I(x))$

- Premise 2: John is a student of this class.

Then *Premise 2* says: $C(John)$

- And the *Conclusion* says: $I(John)$

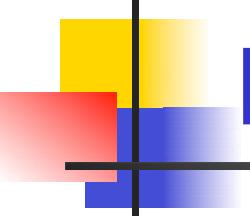
$$\frac{\exists x (C(x) \wedge I(x)) \quad C(John)}{\therefore I(John)}$$



Answer (cont'd)

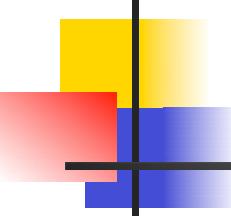
$$\frac{\exists x (C(x) \wedge I(x))}{\begin{array}{c} C(John) \\ \hline \therefore I(John) \end{array}}$$

- No, the argument is invalid; we can disprove it with a counter-example, as follows:
- Consider a case where there is only one intelligent student A in the class, and $A \neq John$.
- Then by existential instantiation of the premise $\exists x (C(x) \wedge I(x))$, $C(A) \wedge I(A)$ is true,
- But the conclusion $I(John)$ is false, since A is the only intelligent student in the class, and $John \neq A$.
- Therefore, the premises *do not* imply the conclusion.



More Proof Examples

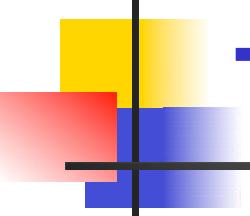
- Is this argument correct or incorrect?
 - “All TAs compose easy quizzes.
Mike is a TA.
Therefore, Mike composes easy quizzes.”
-
- First, separate the premises from conclusion:
 - *Premise 1: All TAs compose easy quizzes.*
 - *Premise 2: Mike is a TA.*
 - *Conclusion: Mike composes easy quizzes.*



Answer

- Next, re-render the example in logic notation.
 - Premise 1: All TAs compose easy quizzes.
 - Let the domain = all people
 - Let $T(x)$ = “ x is a TA”
 - Let $E(x)$ = “ x composes easy quizzes”
 - Then Premise 1 says: $\forall x(T(x) \rightarrow E(x))$
 - Premise 2: Mike is a TA.
 - Let M = Mike
 - Then Premise 2 says: $T(M)$
 - And the Conclusion says: $E(M)$

$$\frac{\forall x (T(x) \rightarrow E(x)) \quad T(M)}{\therefore E(M)}$$



The Proof in Gory Detail

- The argument is correct, because it can be reduced to a sequence of applications of valid inference rules as follows:

$$\frac{\forall x (T(x) \rightarrow E(x))}{\frac{T(M)}{\therefore E(M)}}$$

- Statement

1. $\forall x (T(x) \rightarrow E(x))$
2. $T(M) \rightarrow E(M)$
3. $T(M)$
4. $E(M)$

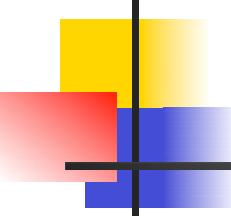
- How obtained

(Premise #1)

(Universal
Instantiation)

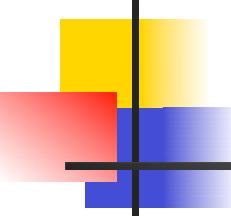
(Premise #2)

(*Modus Ponens* from 6, 19)



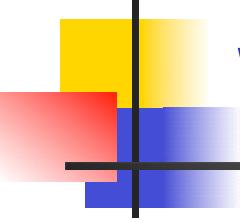
Another Example

- Prove that the sum of a rational number and an irrational number is always irrational.
- First, you have to understand exactly what the question is asking you to prove:
- “For all real numbers x, y ,
if x is rational and y is irrational, then $x+y$ is
irrational.”
- $\forall x, y: \text{Rational}(x) \wedge \text{Irrational}(y) \rightarrow \text{Irrational}(x+y)$



Answer

- Next, think back to the definitions of the terms used in the statement of the theorem:
 - \forall reals r : $\text{Rational}(r) \leftrightarrow \exists \text{ Integer}(i) \wedge \text{Integer}(j \text{ with } j \neq 0): r = i/j.$
 - \forall reals r : $\text{Irrational}(r) \leftrightarrow \neg \text{Rational}(r)$
- You almost always need the definitions of the terms in order to prove the theorem!
- Next, let's go through one valid proof:



What you might write

■ ■ **Theorem:**

$\forall x, y : \text{Rational}(x) \wedge \text{Irrational}(y) \rightarrow \text{Irrational}(x+y)$

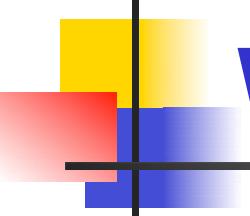
■ ■ **Proof:** Let x, y be any rational and irrational numbers, respectively. ... (universal generalization)

■ ■ Now, just from this, what do we know about x and y ?

Think back to the definition of a rational number:

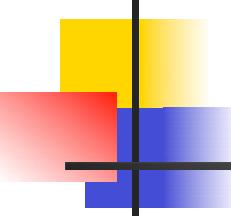
■ ■ ... Since x is rational, we know (from the very definition of rational) that there must be some integers i and j such that $x = i / j$. So, let i_x, j_x be such integers ...

■ ■ Notice that gave them the unique names i_x and j_x so we can refer to them later.



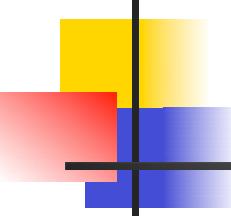
What next?

- What do we know about y ? Only that y is irrational: $\neg\exists$ integers $i,j: y = i/j$.
- But, it's difficult to see how to use a direct proof in this case. So let's try to use proof by contradiction.
- So, what are we trying to show?
Just that $x+y$ is irrational. That is, $\neg\exists i,j: (x + y) = i/j$.
- Now we need to hypothesize the negation of this statement!



More writing...

- Suppose that $x+y$ were not irrational.
Then $x + y$ would be rational, so \exists integers i, j : $x + y = i/j$. So, let i_s and j_s be any such integers where $x + y = i_s/j_s$.
- Now, with all these things named, we can see what happens when we put them together.
- So, we have that $(i_x/j_x) + y = (i_s/j_s)$.
- Notice: We have enough information now to conclude something useful about y , by solving this equation for it!

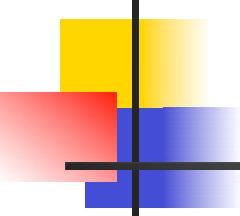


Finishing the Proof

- Solving that equation for y , we have:

$$\begin{aligned}y &= (i_s/j_s) - (i_x/j_x) \\&= (i_s j_x - i_x j_s)/(j_s j_x)\end{aligned}$$

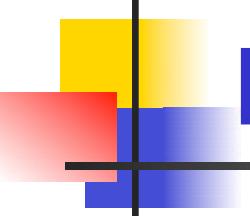
- Now, since the numerator and denominator of this expression are both integers, y is rational (by definition of a rational number).
- This contradicts the assumption that y is irrational. Therefore, our hypothesis that $x+y$ is rational must be false, and so the theorem is proved.



Example of a Wrong Answer

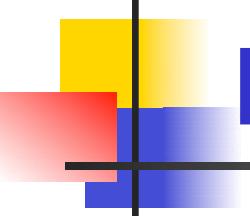
- 1 is rational. $\sqrt{2}$ is irrational. $1+\sqrt{2}$ is irrational.
Therefore, the sum of a rational number and an irrational number is irrational.
(Attempting a direct proof.)

- Why does this answer deserve no credit?
 - We attempted to use an example to prove a universal statement.
- This is always invalid!**
- Even as an example, it's incomplete, because we never even proved that $1+\sqrt{2}$ is irrational!



Proof Terminology

- A ***proof*** is a valid argument that establishes the truth of a mathematical statement
- ***Axiom*** (or ***postulate***): a statement that is assumed to be true
- ***Theorem***
 - A statement that has been proven to be true
- ***Hypothesis, premise***
 - An assumption (often unproven) defining the structures about which we are reasoning



More Proof Terminology

■ ***Lemma***

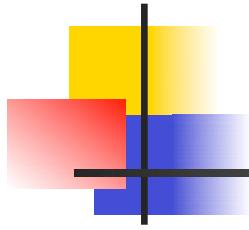
- A minor theorem used as a stepping-stone to proving a major theorem.

■ ***Corollary***

- A minor theorem proved as an easy consequence of a major theorem.

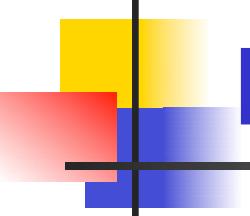
■ ***Conjecture***

- A statement whose truth value has not been proven. (A conjecture may be widely believed to be true, regardless.)



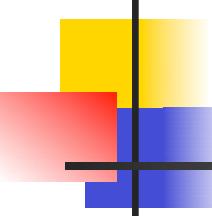
Proof Methods

- For proving a statement p alone
 - ***Proof by Contradiction*** (indirect proof):
Assume $\neg p$, and prove $\neg p \rightarrow F$.



Proof Methods

- For proving implications $p \rightarrow q$, we have:
 - **Trivial proof:** Prove q by itself.
 - **Direct proof:** Assume p is true, and prove q .
 - **Indirect proof:**
 - **Proof by Contraposition** ($\neg q \rightarrow \neg p$):
Assume $\neg q$, and prove $\neg p$.
 - **Proof by Contradiction:**
Assume $p \wedge \neg q$, and show this leads to a contradiction. (i.e. prove $(p \wedge \neg q) \rightarrow F$)
 - **Vacuous proof:** Prove $\neg p$ by itself.



Direct Proof Example

- **Definition:** An integer n is called *odd* iff $n=2k+1$ for some integer k ; n is *even* iff $n=2k$ for some k .
- **Theorem:** Every integer is either odd or even, but not both.
 - This can be proven from even simpler axioms.

- **Theorem:**

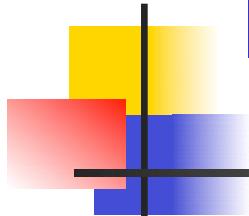
(For all integers n) If n is odd, then n^2 is odd.

Proof:

If n is odd, then $n = 2k + 1$ for some integer k .

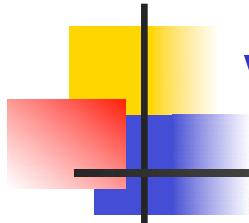
$$\text{Thus, } n^2 = (2k + 1)^2 = 4k^2 + 4k + 1 = 2(2k^2 + 2k) + 1.$$

Therefore n^2 is of the form $2j + 1$ (with j the integer $2k^2 + 2k$), thus n^2 is odd. ■



Indirect Proof Example: Proof by Contraposition

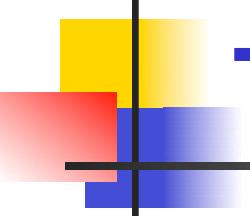
- **Theorem:** (For all integers n)
If $3n + 2$ is odd, then n is odd.
- **Proof:**
(Contrapositive: If n is even, then $3n + 2$ is even)
Suppose that the conclusion is false, i.e., that n is even.
Then $n = 2k$ for some integer k .
Then $3n + 2 = 3(2k) + 2 = 6k + 2 = 2(3k + 1)$.
Thus $3n + 2$ is even, because it equals $2j$ for an integer
 $j = 3k + 1$. So $3n + 2$ is not odd.
We have shown that $\neg(n \text{ is odd}) \rightarrow \neg(3n + 2 \text{ is odd})$,
thus its contrapositive $(3n + 2 \text{ is odd}) \rightarrow (n \text{ is odd})$ is
also true. ■



Vacuous Proof Example

- Show $\neg p$ (i.e. p is false) to prove $p \rightarrow q$ is true.
- **Theorem:** (For all n) If n is both odd and even, then $n^2 = n + n$.
- **Proof:**

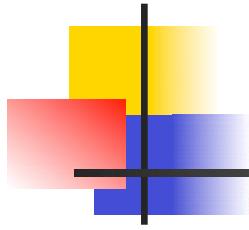
The statement “ n is both odd and even” is necessarily false, since no number can be both odd and even. So, the theorem is vacuously true. ■



Trivial Proof Example

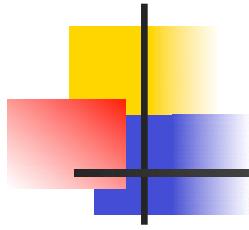
- Show q (i.e. q is true) to prove $p \rightarrow q$ is true.
- **Theorem:** (For integers n) If n is the sum of two prime numbers, then either n is odd or n is even.
- **Proof:**

Any integer n is either odd or even. So the conclusion of the implication is true regardless of the truth of the hypothesis. Thus the implication is true trivially. ■



Proof by Contradiction

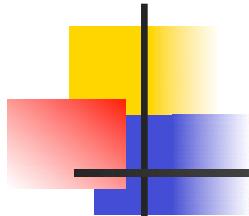
- A method for proving p .
 - Assume $\neg p$, and prove both q and $\neg q$ for some proposition q . (Can be anything!)
 - Thus $\neg p \rightarrow (q \wedge \neg q)$
 - $(q \wedge \neg q)$ is a trivial contradiction, equal to F
 - Thus $\neg p \rightarrow F$, which is only true if $\neg p = F$
 - Thus p is true



Rational Number

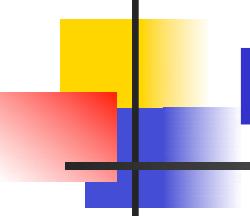
- Definition:

The real number r is *rational* if there exist integers p and q with $q \neq 0$ such that $r = p/q$.
A real number that is not rational is called *irrational*.



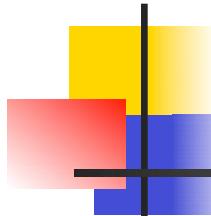
Proof by Contradiction Example

- **Theorem:** $\sqrt{2}$ is irrational.
 - **Proof:**
 - Assume that $\sqrt{2}$ is rational. This means there are integers x and y ($y \neq 0$) with no common divisors such that $\sqrt{2} = x/y$.
Squaring both sides, $2 = x^2/y^2$, so $2y^2 = x^2$. So x^2 is even; thus x is even (see earlier).
Let $x = 2k$. So $2y^2 = (2k)^2 = 4k^2$. Dividing both sides by 2, $y^2 = 2k^2$. Thus y^2 is even, so y is even.
But then x and y have a common divisor, namely 2, so we have a contradiction.
Therefore, $\sqrt{2}$ is irrational. ■



Proof by Contradiction

- Proving implication $p \rightarrow q$ by contradiction
 - Assume $\neg q$, and use the premise p to arrive at a contradiction, i.e. $(\neg q \wedge p) \rightarrow \mathbf{F}$
 $(p \rightarrow q \equiv (\neg q \wedge p) \rightarrow \mathbf{F})$
 - How does this relate to the proof by contraposition?
 - ***Proof by Contraposition*** ($\neg q \rightarrow \neg p$):
Assume $\neg q$, and prove $\neg p$.



Proof by Contradiction

Example: Implication

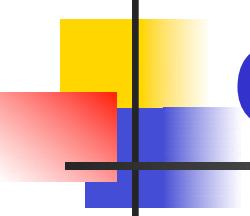
- **Theorem:** (For all integers n)
If $3n + 2$ is odd, then n is odd.
- **Proof:**

Assume that the conclusion is false, i.e., that n is even, and that $3n + 2$ is odd.

Then $n = 2k$ for some integer k and $3n + 2 = 3(2k) + 2 = 6k + 2 = 2(3k + 1)$. Thus $3n + 2$ is even, because it equals $2j$ for an integer $j = 3k + 1$.

This contradicts the assumption “ $3n + 2$ is odd”.

This completes the proof by contradiction, proving that if $3n + 2$ is odd, then n is odd. ■



Circular Reasoning

- The fallacy of (explicitly or implicitly) assuming the very statement you are trying to prove in the course of its proof. Example:
- Prove that an integer n is even, if n^2 is even.
- **Attempted proof:**

Assume n^2 is even. Then $n^2 = 2k$ for some integer k .

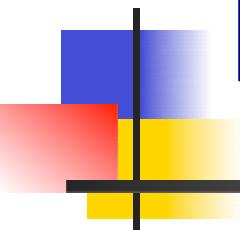
Dividing both sides by n gives $n = (2k)/n = 2(k/n)$.

So there is an integer j (namely k/n) such that $n = 2j$.
Therefore n is even.

- Circular reasoning is used in this proof.

Where?

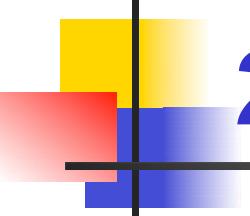
*Begs the question: How do
you show that $j = k/n = n/2$ is an integer,
without first assumig that n is even?*



Lecture 8

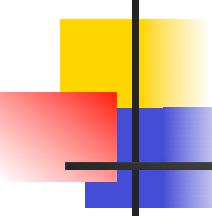
Chapter 2. Basic Structures

1. Sets
2. Set Operations



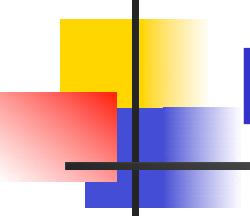
2.1 Sets

- A **set** is a new type of structure, representing an ***unordered*** collection (group) of zero or more ***distinct*** (different) objects. The objects are called ***elements*** or ***members*** of the set.
- Notation: $x \in S$
- Set theory deals with operations between, relations among, and statements about sets.
- Sets are ubiquitous in computer software systems.
- (E.g. data types `Set`, `HashSet` in `java.util`)



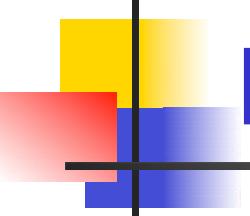
Basic Notations for Sets

- For sets, we'll use variables S, T, U, \dots
- We can denote a set S in writing by listing all of its elements in curly braces:
- $\{a, b, c\}$ is the set whose elements are a, b , and c
- ***Set builder notation:***
- For any statement $P(x)$ over any domain, $\{x \mid P(x)\}$ is *the set of all x such that $P(x)$ is true*
- Example: $\{1, 2, 3, 4\}$
= $\{x \mid x \text{ is an integer where } x > 0 \text{ and } x < 5\}$
= $\{x \in \mathbf{Z} \mid x > 0 \text{ and } x < 5\}$



Basic Properties of Sets

- Sets are inherently ***unordered***:
 - No matter what objects a , b , and c denote,
 $\{a, b, c\} = \{a, c, b\} = \{b, a, c\} =$
 $\{b, c, a\} = \{c, a, b\} = \{c, b, a\}.$
 - All elements are ***distinct*** (unequal); multiple listings make no difference!
 - If $a = b$, then $\{a, b, c\} = \{a, c\} = \{b, c\} =$
 $\{a, a, b, a, b, c, c, c\}.$
 - This set contains (at most) 2 elements!



Definition of Set Equality

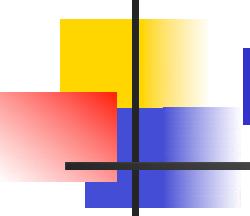
- Two sets are declared to be equal *if and only if* they contain exactly the same elements.
- In particular, it does not matter *how the set is defined or denoted.*

- Example:

The set {1, 2, 3, 4}

= { $x \mid x$ is an integer where $x > 0$ and $x < 5$ }

= { $x \mid x$ is a positive integer where $x^2 < 20$ }



Infinite Sets

- Conceptually, sets may be *infinite* (i.e., not *finite*, without end, unending).
 - Symbols for some special infinite sets:

N = {0, 1, 2,...} the set of **Natural numbers**.

Z = {..., -2, -1, 0, 1, 2,...} the set of

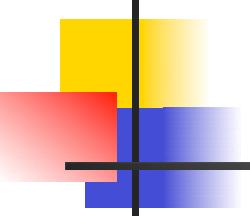
Zntegers. **Z**⁺ = {1, 2, 3,...} the set of positive integers.

Q = { $p/q \mid p,q \in \mathbf{Z}$, and $q \neq 0$ }

the set of Rational numbers.

R = the set of “Real” numbers.

- “Blackboard Bold” or double-struck font is also often used for these special number sets.



Basic Set Relations

- $x \in S$ (“ x is in S ”) is the proposition that object x is an *element* or *member* of set S .

■■ e.g. $3 \in \mathbb{N}$,

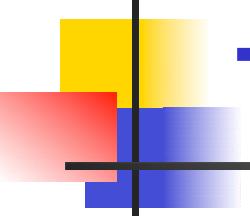
$a \in \{x \mid x \text{ is a letter of the alphabet}\}$

■■ Can define set equality in terms of \in relation:

$$\forall S, T: S = T \leftrightarrow [\forall x (x \in S \leftrightarrow x \in T)]$$

“Two sets are equal iff they have all the same members.”

- $x \notin S \equiv \neg(x \in S)$ “ x is not in S ”

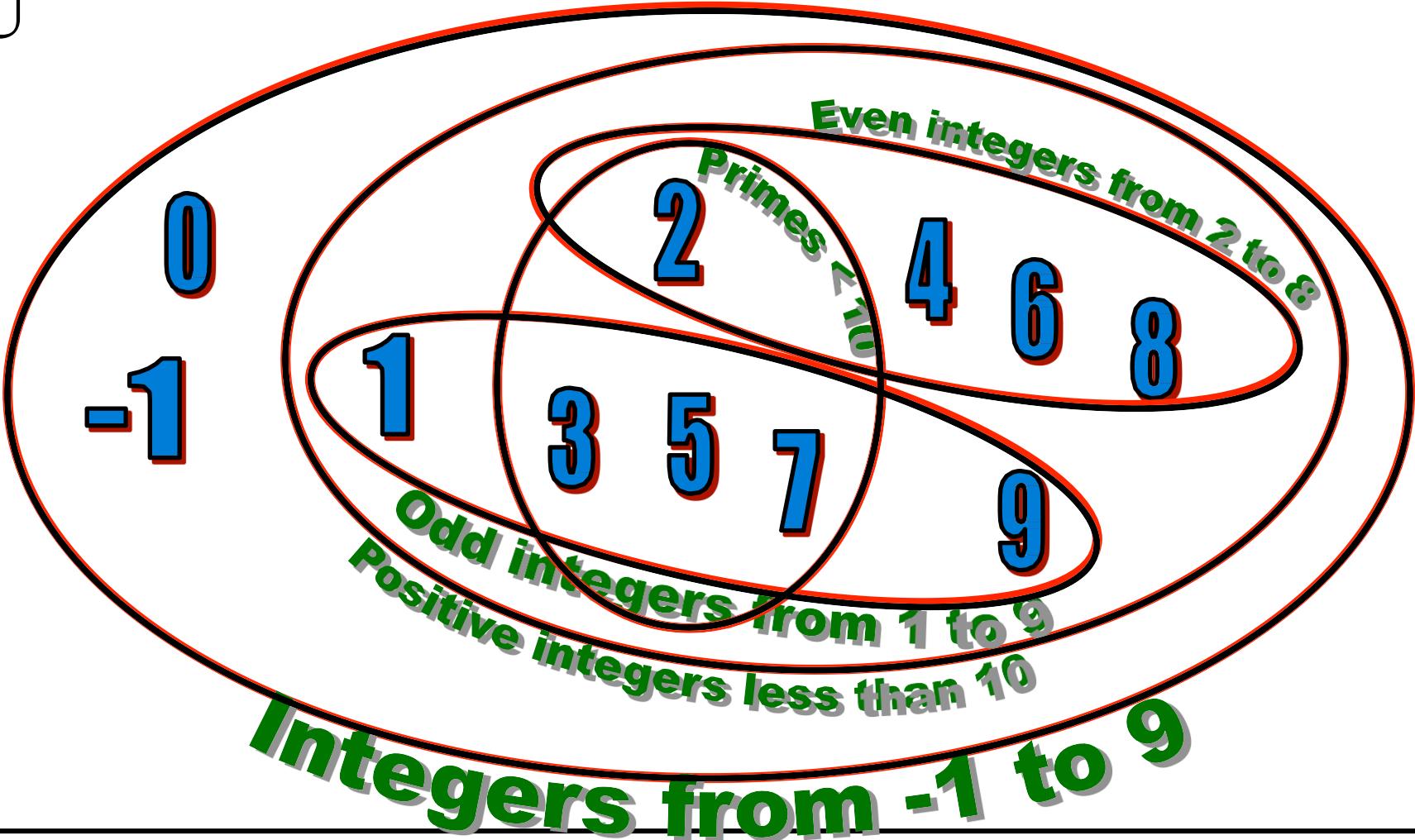


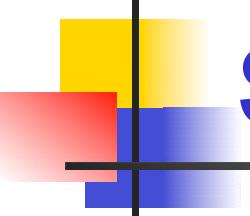
The Empty Set

- \emptyset (“null”, “the empty set”) is the unique set that contains no elements whatsoever.
- $\emptyset = \{ \} = \{x \mid \text{False}\}$
- No matter the domain of discourse, we have the axiom $\neg \exists x: x \in \emptyset$.
- $\{ \} \neq \{\emptyset\} = \{ \{ \} \}$
- $\{\emptyset\}$ it isn’t empty because it has \emptyset as a member!

Venn Diagrams

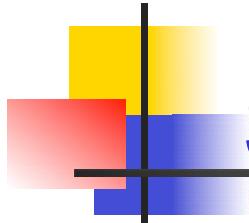
U





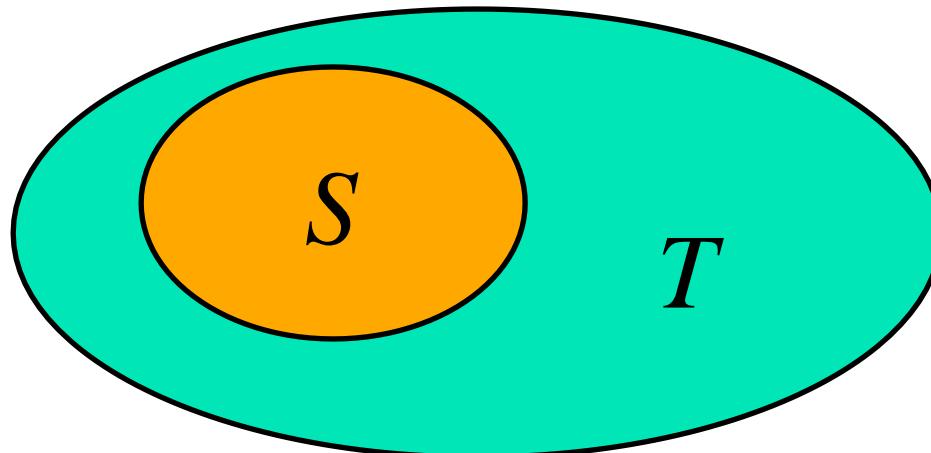
Subset and Superset

- $S \subseteq T$ (“ S is a subset of T ”) means that every element of S is also an element of T .
- $S \subseteq T \equiv \forall x (x \in S \rightarrow x \in T)$
- $\emptyset \subseteq S, S \subseteq S$
- $S \subseteq T$ (“ S is a superset of T ”) means $T \subseteq S$
- Note $(S = T) \equiv (S \subseteq T \wedge S \supseteq T)$
 $\equiv \forall x(x \in S \rightarrow x \in T) \wedge \forall x(x \in T \rightarrow x \in S)$
 $\equiv \forall x(x \in S \leftrightarrow x \in T)$
- $S \not\subseteq T$ means $\neg(S \subseteq T)$, i.e. $\exists x(x \in S \wedge x \notin T)$

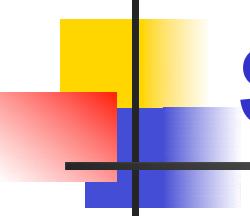


Proper (Strict) Subsets & Supersets

- $S \subset T$ (“ S is a proper subset of T ”) means that $S \subseteq T$ but $T \not\subseteq S$. Similar for $S \supset T$.
- Example:
 $\{1, 2\} \subset \{1, 2, 3\}$



Venn Diagram of $S \subset T$



Sets Are Objects, Too!

■■ The objects that are elements of a set may *themselves* be sets.

■■ Example:

Let $S = \{x \mid x \subseteq \{1, 2, 3\}\}$ then $S = \{\emptyset,$
 $\{1\}, \{2\}, \{3\},$
 $\{1, 2\}, \{1, 3\}, \{2, 3\},$
 $\{1, 2, 3\}\}$

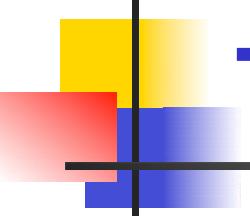
■■ Note that $1 \neq \{1\} \neq \{\{1\}\}$!!!!



Cardinality and Finiteness

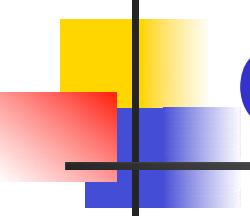
- $|S|$ (read “the *cardinality* of S ”) is a measure of how many different elements S has.
- E.g., $|\emptyset| = 0$, $|\{1, 2, 3\}| = 3$, $|\{a, b\}| = 2$,
- 
 $|\{\{1, 2, 3\}, \{4, 5\}\}| = \underline{\hspace{2cm}}$
- If $|S| \in \mathbf{N}$, then we say S is *finite*. Otherwise, we say S is *infinite*.
- What are some infinite sets we’ve seen?

N, Z, Q, R



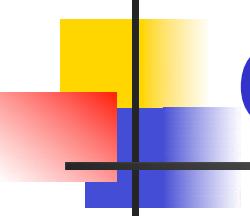
The *Power Set* Operation

- The **power set** $P(S)$ of a set S is the set of all subsets of S . $P(S) = \{x \mid x \subseteq S\}$.
- Examples
 - $P(\{a, b\}) = \{ \emptyset, \{a\}, \{b\}, \{a, b\} \}$
 - $S = \{0, 1, 2\}$
 $P(S) = \{\emptyset, \{0\}, \{1\}, \{2\}, \{0, 1\}, \{0, 2\}, \{1, 2\}, \{0, 1, 2\}\}$
 - $P(\emptyset) = \{\emptyset\}$
 - $P(\{\emptyset\}) = \{\emptyset, \{\emptyset\}\}$
- Note that for finite S , $|P(S)| = 2^{|S|}$.
- It turns out $\forall S (|P(S)| > |S|)$, e.g. $|P(\mathbb{N})| > |\mathbb{N}|$.



Ordered n -tuples

- These are like sets, except that duplicates matter, and the order makes a difference.
- For $n \in \mathbb{N}$, an *ordered n -tuple* or a *sequence* or *list of length n* is written (a_1, a_2, \dots, a_n) . Its *first* element is a_1 , its *second* element is a_2 , etc.
- Note that $(1, 2) \neq (2, 1) \neq (2, 1, 1)$.Contrast with sets' {}
- Empty sequence, singlets, pairs, triples, quadruples, quintuples, ..., n -tuples.



Cartesian Products of Sets

- For sets A and B , their **Cartesian product** denoted by $A \times B$, is the set of all ordered pairs (a, b) , where $a \in A$ and $b \in B$. Hence,

$$A \times B = \{ (a, b) \mid a \in A \wedge b \in B \}.$$

- E.g. $\{a, b\} \times \{1, 2\}$

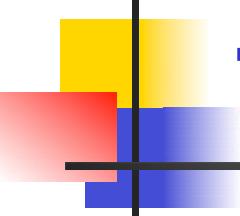
$$= \{ (a, 1), (a, 2), (b, 1), (b, 2) \}$$

- Note that for finite A, B , $|A \times B| = |A||B|$.

- Note that the Cartesian product is **not** commutative: i.e., $\neg \forall A, B (A \times B = B \times A)$.

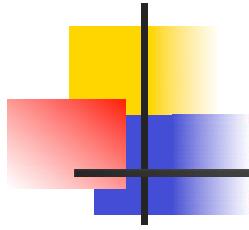
- Extends to $A_1 \times A_2 \times \dots \times A_n$

$$= \{ (a_1, a_2, \dots, a_n) \mid a_i \in A_i \text{ for } i = 1, 2, \dots, n \}$$



The Union Operator

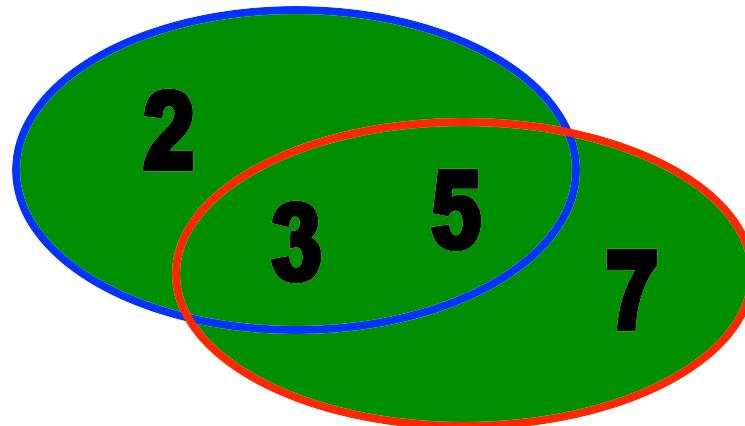
- For sets A and B , their ***union*** $A \cup B$ is the set containing all elements that are either in A , or (“ \vee ”) in B (or, of course, in both).
- Formally, $\forall A, B: A \cup B = \{x \mid x \in A \vee x \in B\}$.
- Note that $A \cup B$ is a **superset** of both A and B (in fact, it is the smallest such superset):
 $\forall A, B: (A \cup B \subseteq A) \wedge (A \cup B \subseteq B)$

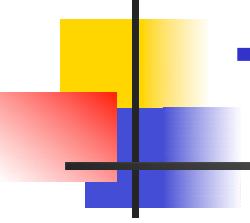


Union Examples

- $\{a, b, c\} \cup \{2, 3\} = \{a, b, c, 2, 3\}$
- $\{2, 3, 5\} \cup \{3, 5, 7\} = \{2, 3, 5, 3, 5, 7\}$
= $\{2, 3, 5, 7\}$

Required Form

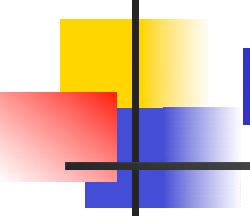




The Intersection Operator

- For sets A and B , their ***intersection*** $A \cap B$ is the set containing all elements that are simultaneously in A **and** (“ \wedge ”) in B .
- Formally, $\forall A, B: A \cap B = \{x \mid x \in A \wedge x \in B\}$.
- Note that $A \cap B$ is a **subset** of both A and B (in fact it is the largest such subset):

$$\forall A, B: (A \cap B \subseteq A) \wedge (A \cap B \subseteq B)$$

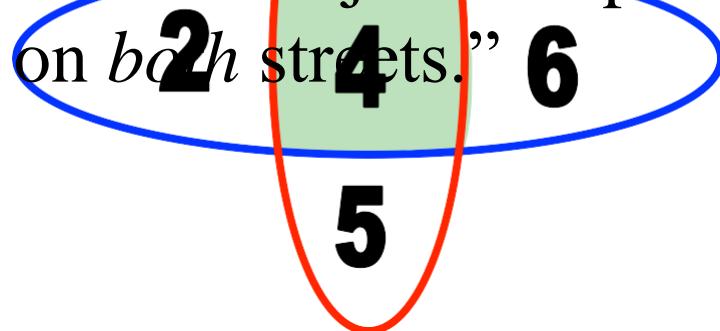


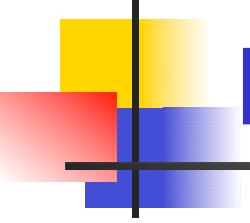
Intersection Examples

■■ $\{a, b, c\} \cap \{2, 3\} = \underline{\emptyset}$

■■ $\{2, 4, 6\} \cap \{3, 4, 5\} = \underline{\{4\}}$

Think “The 3ntersection of University Ave. and Dole St. is just that part of the road surface that lies on *both* streets.”

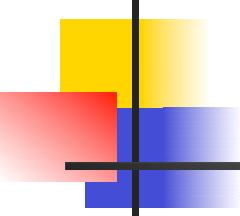




Disjointedness

- Two sets A, B are called ***disjoint*** (i.e., unjoined) iff their intersection is empty. ($A \cap B = \emptyset$)
- Example: the set of even integers is disjoint with the set of odd integers.





Inclusion-Exclusion Principle

- How many elements are in $A \cup B$?

$$|A \cup B| = |A| + |B| - |A \cap B|$$

- Example: How many students in the class major in Computer Science or Mathematics?

- Consider set $E = C \cup M$,

$C = \{s \mid s \text{ is a Computer Science major}\}$

$M = \{s \mid s \text{ is a Mathematics major}\}$

- Some students are joint majors!

$$|E| = |C \cup M| = |C| + |M| - |C \cap M|$$

Inclusion-Exclusion Principle

- How many elements are in $A \cup B$?

$$|A \cup B| = |A| + |B| - |A \cap B|$$

- Example: How many students in the class major in Computer Science or Mathematics?

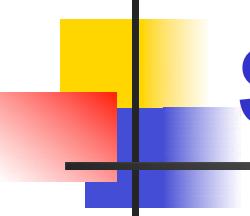
- Consider set $E = C \cup M$,

$C = \{s \mid s \text{ is a Computer Science major}\}$

$M = \{s \mid s \text{ is a Mathematics major}\}$

- Some students are joint majors!

$$|E| = |C \cup M| = |C| + |M| - |C \cap M|$$

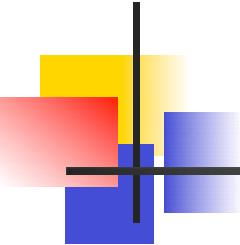


Set Difference

- For sets A and B , the ***difference of A and B***, written $A - B$, is the set of all elements that are in A but not B .
- Formally:

$$\begin{aligned}A - B &= \{x \mid x \in A \wedge x \notin B\} \\&= \{x \mid \neg(x \in A \rightarrow x \in B)\}\end{aligned}$$

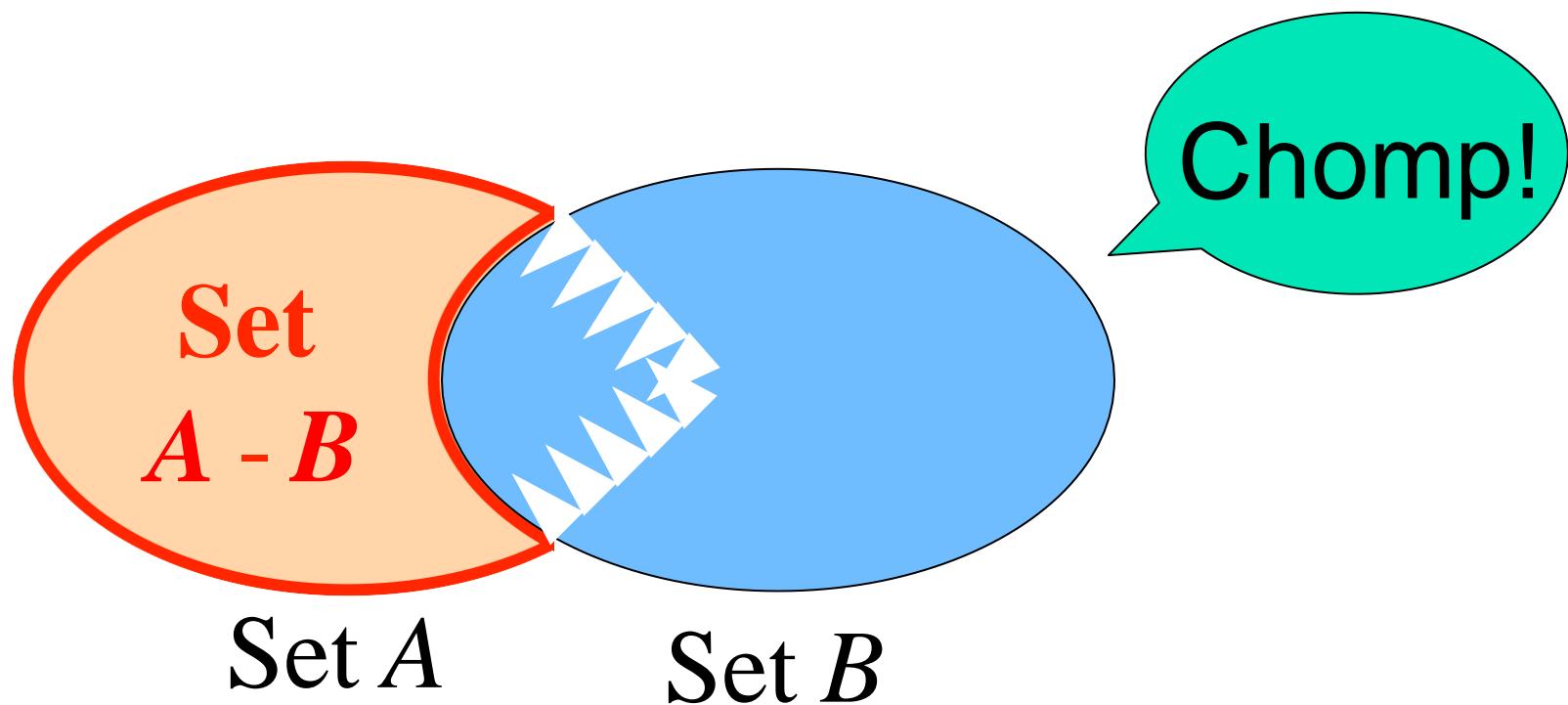
- Also called:
The ***complement of B with respect to A***.



Set Difference: Venn Diagram

■■ $A - B$

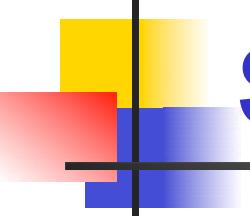
is what's left after B “takes a bite out of A ”



Set Difference Examples

$$\begin{aligned} & \text{---} \\ & - \{1, 2, 3, 4, 5, 6\} - \{2, 3, 5, 7, 9, 11\} = \\ & \quad \underline{\{1, 4, 6\}} \end{aligned}$$

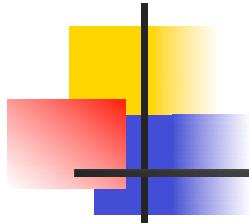
$$\begin{aligned} & \text{---} \\ & - \mathbf{Z} - \mathbf{N} = \{\dots, -1, 0, 1, 2, \dots\} - \{0, 1, \dots\} \\ & = \{x \mid x \text{ is an integer but not a natural \#}\} \\ & = \{\dots, -3, -2, -1\} \\ & = \{x \mid x \text{ is a negative integer}\} \end{aligned}$$



Set Complements

- The *universe of discourse* (or the *domain*) can itself be considered a set, call it U .
- When the context clearly defines U , we say that for any set $A \subseteq U$, the **complement** of A , written as \overline{A} , is the complement of A with respect to U , i.e., it is $U - A$.
- E.g., If $U = \mathbb{N}$,

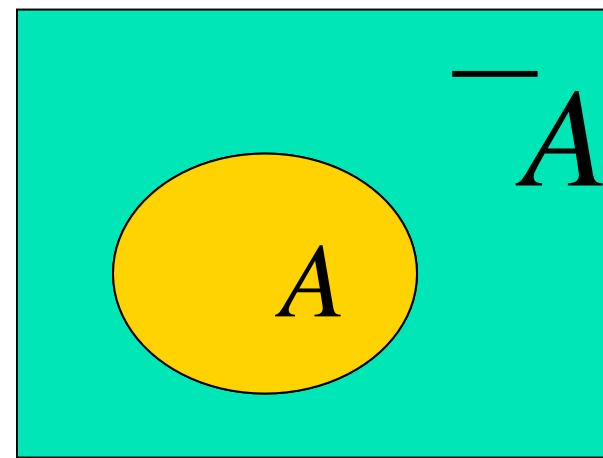
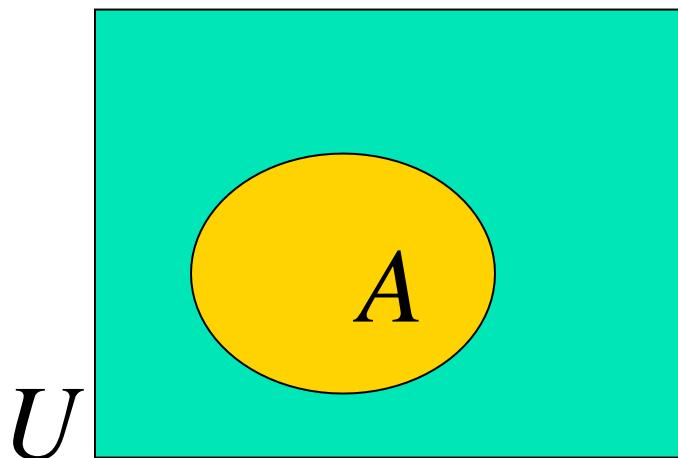
$$\overline{\{3, 5\}} = \{0, 1, 2, 4, 6, 7, \dots\}$$

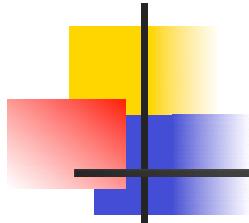


More on Set Complements

■■ An equivalent definition, when U is obvious:

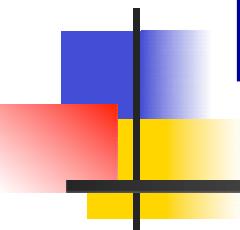
$$\overline{A} = \{x \mid x \notin A\}$$





Interval Notation

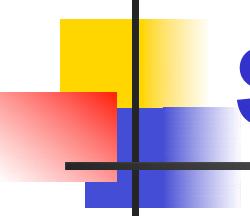
- $a, b \in \mathbb{R}$, and $a < b$ then
 - $(a, b) = \{x \in \mathbb{R} \mid a < x < b\}$
 - $[a, b] = \{x \in \mathbb{R} \mid a \leq x \leq b\}$
 - $(a, b] = \{x \in \mathbb{R} \mid a < x \leq b\}$
 - $(-\infty, b] = \{x \in \mathbb{R} \mid x \leq b\}$
 - $[a, \infty) = \{x \in \mathbb{R} \mid a \leq x\}$
 - $(a, \infty) = \{x \in \mathbb{R} \mid a < x\}$



Lecture 9

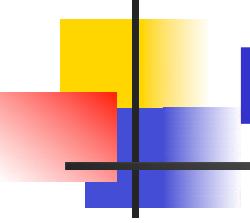
Chapter 2. Basic Structures

2.2 Set Operations



Set Identities

- Identity: $A \cup \emptyset = A = A \cap U$
- Domination: $A \cup U = U, A \cap \emptyset = \emptyset$
- Idempotent: $A \cup A = A, A \cap A = A$
- Double complement: $\overline{\overline{A}} = A$
- Commutative: $A \cup B = B \cup A, A \cap B = B \cap A$
- Associative: $A \cup (B \cup C) = (A \cup B) \cup C,$
 $A \cap (B \cap C) = (A \cap B) \cap C$
- Distributive: $A \cap (B \cup C) = (A \cap B) \cup (A \cap C),$
 $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
- Absorption: $A \cup (A \cap B) = A, A \cap (A \cup B) = A$
- Complement: $A \cup \overline{A} = U, A \cap \overline{A} = \emptyset$

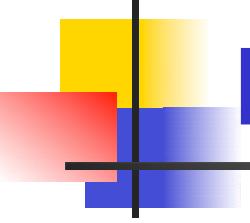


DeMorgan's Law for Sets

- Exactly analogous to (and provable from) DeMorgan's Law for propositions.

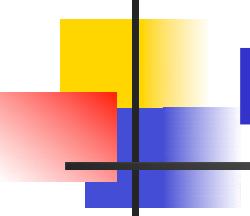
$$\overline{A \cup B} = \overline{A} \cap \overline{B}$$

$$\overline{A \cap B} = \overline{A} \cup \overline{B}$$



Proving Set Identities

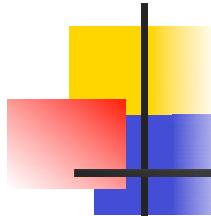
- To prove statements about sets, of the form $E_1 = E_2$ (where the E s are set expressions), here are three useful techniques:
 1. Prove $E_1 \subseteq E_2$ and $E_2 \subseteq E_1$ separately.
 2. Use set builder notation & logical equivalences.
 3. Use a *membership table*.
 4. Use a Venn diagram.



Method 1: Mutual Subsets

Example: Show $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$.

- Part 1: Show $A \cap (B \cup C) \subseteq (A \cap B) \cup (A \cap C)$.
 - Assume $x \in A \cap (B \cup C)$, & show $x \in (A \cap B) \cup (A \cap C)$.
 - We know that $x \in A$, and either $x \in B$ or $x \in C$.
 - Case 1: $x \in A$ and $x \in B$. Then $x \in A \cap B$,
so $x \in (A \cap B) \cup (A \cap C)$.
 - Case 2: $x \in A$ and $x \in C$. Then $x \in A \cap C$,
so $x \in (A \cap B) \cup (A \cap C)$.
 - Therefore, $x \in (A \cap B) \cup (A \cap C)$.
 - Therefore, $A \cap (B \cup C) \subseteq (A \cap B) \cup (A \cap C)$.
- Part 2: Show $(A \cap B) \cup (A \cap C) \subseteq A \cap (B \cup C)$. (Try it!)



Method 2: Set BuilderNotation & Logical Equivalence

■ Show $\overline{A \cap B} = \overline{A} \cup \overline{B}$

$$\overline{A \cap B} = \{x \mid x \notin (A \cap B)\}$$

def. of complement

$$= \{x \mid \neg(x \in (A \cap B))\}$$

def. of “does not belong”

$$= \{x \mid \neg(x \in A \wedge x \in B)\}$$

def. of intersection

$$= \{x \mid \neg(x \in A) \vee \neg(x \in B)\}$$

De Morgan’s law (logic)

$$= \{x \mid x \notin A \vee x \notin B\}$$

def. of “does not belong”

$$= \{x \mid x \in \overline{A} \vee x \in \overline{B}\}$$

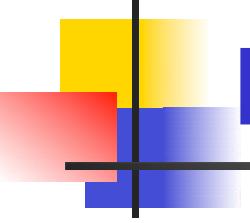
def. of complement

$$= \{x \mid x \in \overline{A} \cup \overline{B}\}$$

def. of union

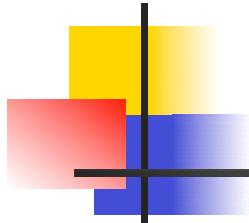
$$= \overline{A} \cup \overline{B}$$

by set builder notation



Method 3: Membership Tables

- Analog to truth tables in propositional logic.
- Columns for different set expressions.
- Rows for all combinations of memberships in constituent sets.
- Use “1” to indicate membership in the derived set, “0” for non-membership.
- Prove equivalence with identical columns.



Membership Table Example

- Prove $(A \cup B) - B = A - B$.

A	B	$A \cup B$	$(A \cup B) - B$	$A - B$
1	1	1	0	0
1	0	1	1	1
0	1	1	0	0
0	0	0	0	0

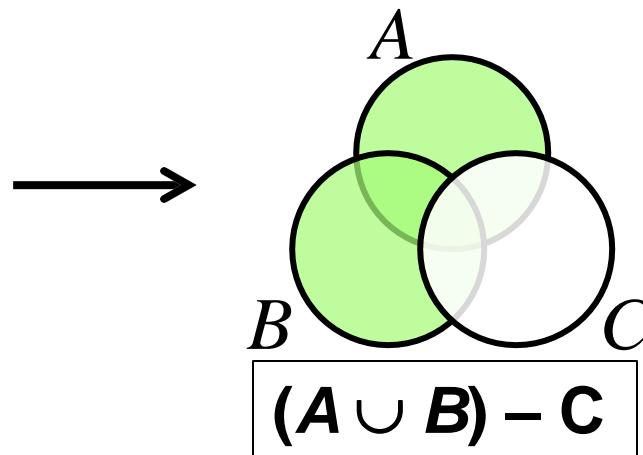
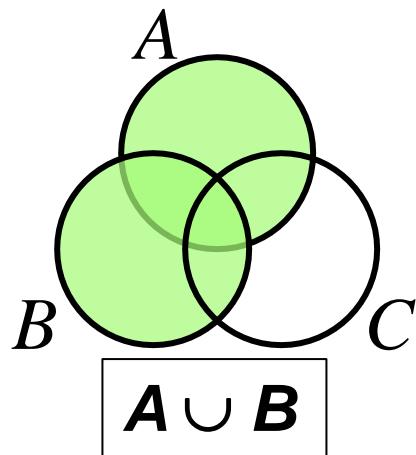
Membership Table Exercise

- Prove $(A \cup B) - C = (A - C) \cup (B - C)$.

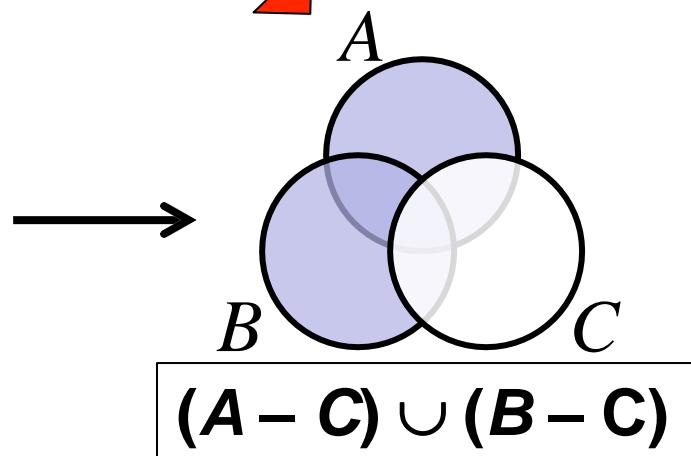
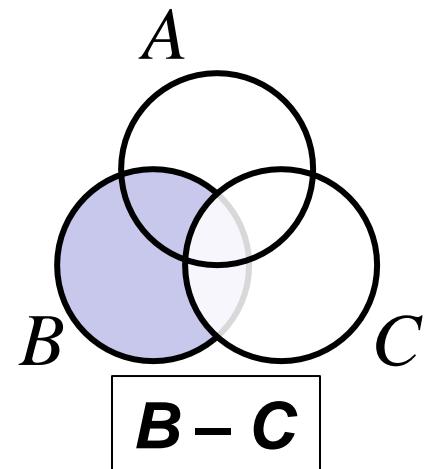
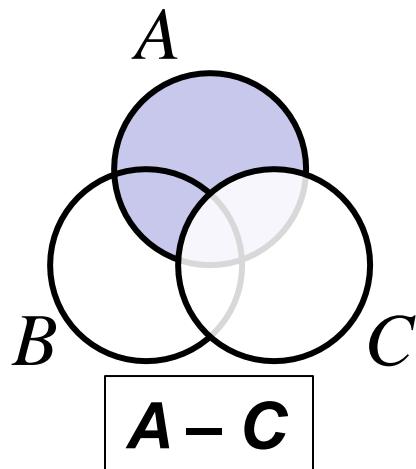
A	B	C	$A \cup B$	$(A \cup B) - C$	$A - C$	$B - C$	$(A - C) \cup (B - C)$
1	1	1	1	0	0	0	0
1	1	0	1	1	1	1	1
1	0	1	1	0	0	0	0
1	0	0	1	1	1	0	1
0	1	1	1	0	0	0	0
0	1	0	1	1	0	1	1
0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0

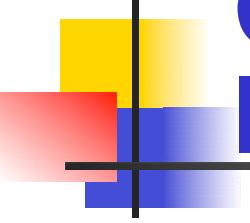
Method 4: Venn Diagram

- Prove $(A \cup B) - C = (A - C) \cup (B - C)$.



SAME!





Generalized Unions & Intersections

- Since union & intersection are commutative and associative, we can extend them from operating on pairs of sets A and B to operating on sequences of sets A_1, \dots, A_n , or even on sets of sets, $X = \{A \mid P(A)\}$.

Generalized Union

- Binary union operator: $A \cup B$

- n -ary union:

$$A_1 \cup A_2 \cup \dots \cup A_n = ((\dots((A_1 \cup A_2) \cup \dots) \cup A_n)$$

(grouping & order is irrelevant)

- “Big U” notation:

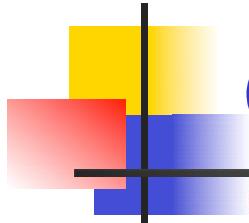
$$\bigcup_{i=1}^n A_i$$

- More generally, union of the sets A_i for $i \in I$:

$$\bigcup_{i \in I} A_i$$

- For infinite number of sets:

$$\bigcup_{i=1}^{\infty} A_i$$



Generalized Union Examples

- Let $A_i = \{i, i+1, i+2, \dots\}$. Then,

$$\bigcup_{i=1}^n A_i = A_1 \cup A_2 \cup A_3 \cup \dots \cup A_n$$

$$\begin{aligned} &= \{1, 2, 3, \dots\} \cup \{2, 3, 4, \dots\} \cup \dots \cup \{n, n+1, n+2, \dots\} \\ &= \{1, 2, 3, \dots\} \end{aligned}$$

- Let $A_i = \{1, 2, 3, \dots, i\}$ for $i = 1, 2, 3, \dots$. Then,

$$\begin{aligned} \bigcup_{i=1}^{\infty} A_i &= A_1 \cup A_2 \cup A_3 \cup \dots \\ &= \{1\} \cup \{1, 2\} \cup \{1, 2, 3\} \cup \dots \\ &= \{1, 2, 3, \dots\} = \mathbb{Z}^+ \end{aligned}$$

Generalized Intersection

- Binary intersection operator: $A \cap B$

- n -ary intersection:

$$A_1 \cap A_2 \cap \dots \cap A_n \equiv ((\dots((A_1 \cap A_2) \cap \dots) \cap A_n)$$

(grouping & order is irrelevant)

- “Big Arch” notation:

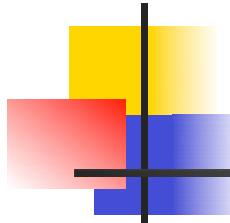
$$\bigcap_{i=1}^n A_i$$

- Generally, intersection of sets A_i for $i \in I$:

$$\bigcap_{i \in I} A_i$$

- For infinite number of sets:

$$\bigcap_{i=1}^{\infty} A_i$$



Generalized Intersection Example

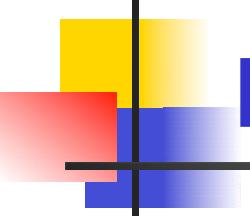
Let $A_i = \{i, i+1, i+2, \dots\}$. Then,

$$\bigcap_{i=1}^n A_i = A_1 \cap A_2 \cap A_3 \cap \cdots \cap A_n$$

$$\begin{aligned} &= \{1, 2, 3, \dots\} \cap \{2, 3, 4, \dots\} \cap \cdots \cap \{n, n+1, n+2, \dots\} \\ &= \{n, n+1, n+2, \dots\} \end{aligned}$$

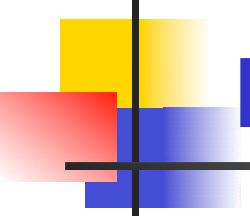
■ Let $A_i = \{1, 2, 3, \dots, i\}$ for $i = 1, 2, 3, \dots$. Then,

$$\begin{aligned} \bigcap_{i=1}^{\infty} A_i &= A_1 \cap A_2 \cap A_3 \cap \cdots \\ &= \{1\} \cap \{1, 2\} \cap \{1, 2, 3\} \cap \cdots \\ &= \{1\} \end{aligned}$$



Bit String Representation of Sets

- A frequent theme of this course are methods of *representing* one discrete structure using another discrete structure of a different type.
- For an enumerable universal set U with ordering x_1, x_2, x_3, \dots , we can represent a finite set $S \subseteq U$ as the finite bit string $B = b_1 b_2 \dots b_n$ where $b_i = 1$ if $x_i \in S$ and $b_i = 0$ if $x_i \notin S$.
- E.g. $U = \mathbb{N}$, $S = \{2, 3, 5, 7, 11\}$, $B = 0011\ 0101\ 0001$.
- In this representation, the set operators “ \cup ”, “ \cap ”, “ $-$ ” are implemented directly by bitwise OR, AND, NOT!



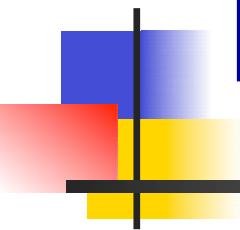
Examples of Sets as Bit Strings

- Let $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$, and the ordering of elements of U has the elements in increasing order, then

$$S_1 = \{1, 2, 3, 4, 5\} \Rightarrow B_1 = 11\ 11100000$$

$$S_2 = \{1, 3, 5, 7, 9\} \Rightarrow B_2 = 10\ 10101010$$

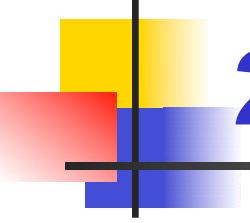
- $S_1 \cup S_2 = \{1, 2, 3, 4, 5, 7, 9\}$
 \Rightarrow bit string = 11 1110 1010 = $B_1 \vee B_2$
- $S_1 \cap S_2 = \{1, 3, 5\}$
 \Rightarrow bit string = 10 1010 0000 = $B_1 \wedge B_2$
- $\overline{S}_1 = \{6, 7, 8, 9, 10\}$
 \Rightarrow bit string = 00 0001 1111 = $\neg B_1$



Lecture 10

Chapter 2. Basic Structures

2.3 Functions

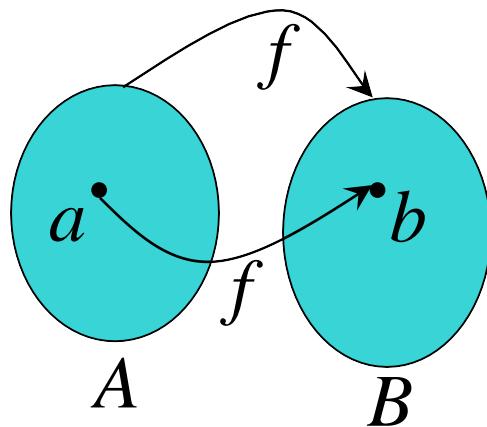


2.3 Functions

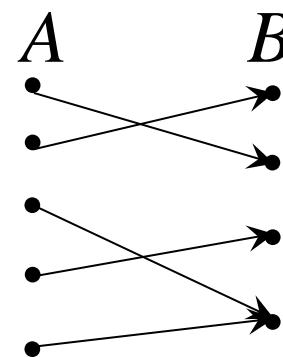
- From calculus, you are familiar with the concept of a real-valued function f , which assigns to each number $x \in \mathbb{R}$ a value $y = f(x)$, where $y \in \mathbb{R}$.
- But, the notion of a function can also be naturally generalized to the concept of assigning elements of *any* set to elements of *any* set. (Also known as a *map*.)

Function: Formal Definition

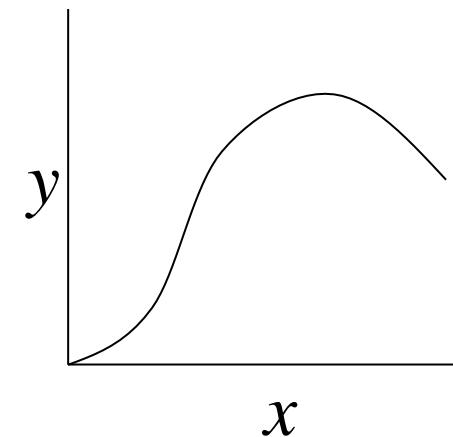
- For any sets A and B , we say that a ***function*** (or “***mapping***”) f from A to B ($f : A \rightarrow B$) is a particular assignment of **exactly one element** $f(x) \in B$ to **each element** $x \in A$.
- Functions can be represented graphically in several ways:



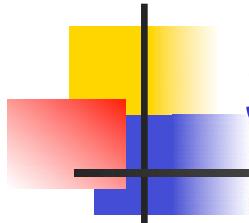
Like Venn diagrams



Bipartite Graph

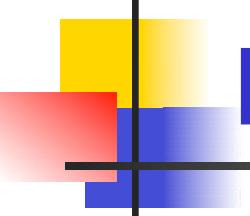


Plot



Some Function Terminology

- If it is written that $f : A \rightarrow B$, and $f(a) = b$ (where $a \in A$ and $b \in B$), then we say:
 - A is the **domain** of f
 - B is the **codomain** of f
 - b is the **image** of a under f
 - a can not have more than 1 image
 - a is a **pre-image** of b under f
 - b may have more than 1 pre-image
 - The **range** $R \subseteq B$ of f is $R = \{b \mid \exists a f(a) = b\}$

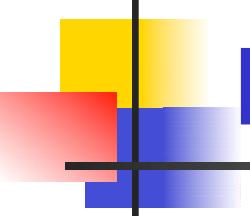


Range versus Codomain

- The range of a function might *not* be its whole codomain.
- The codomain is the set that the function is *declared* to map all domain values into.
- The range is the *particular* set of values in the codomain that the function *actually* maps elements of the domain to.

Range vs. Codomain: Example

- Suppose I declare that: “ f is a function mapping students in this class to the set of grades {A, B, C, D, F}.”
- At this point, you know f ’s codomain is: $\{A, B, C, D, F\}$, and its range is unknown!
- Suppose the grades turn out all As and Bs.
- Then the range of f is $\{A, B\}$, but its codomain is still $\{A, B, C, D, F\}$!



Function Operators

- $+$, \times (“plus”, “times”) are binary operators over \mathbf{R} . (Normal addition & multiplication.)
- Therefore, we can also add and multiply two *real-valued functions* $f, g: \mathbf{R} \rightarrow \mathbf{R}$:
 - $(f + g): \mathbf{R} \rightarrow \mathbf{R}$, where $(f + g)(x) = f(x) + g(x)$
 - $(fg): \mathbf{R} \rightarrow \mathbf{R}$, where $(fg)(x) = f(x)g(x)$
- Example 6:

Let f and g be functions from \mathbf{R} to \mathbf{R} such that $f(x) = x^2$ and $g(x) = x - x^2$. What are the functions $f + g$ and fg ?

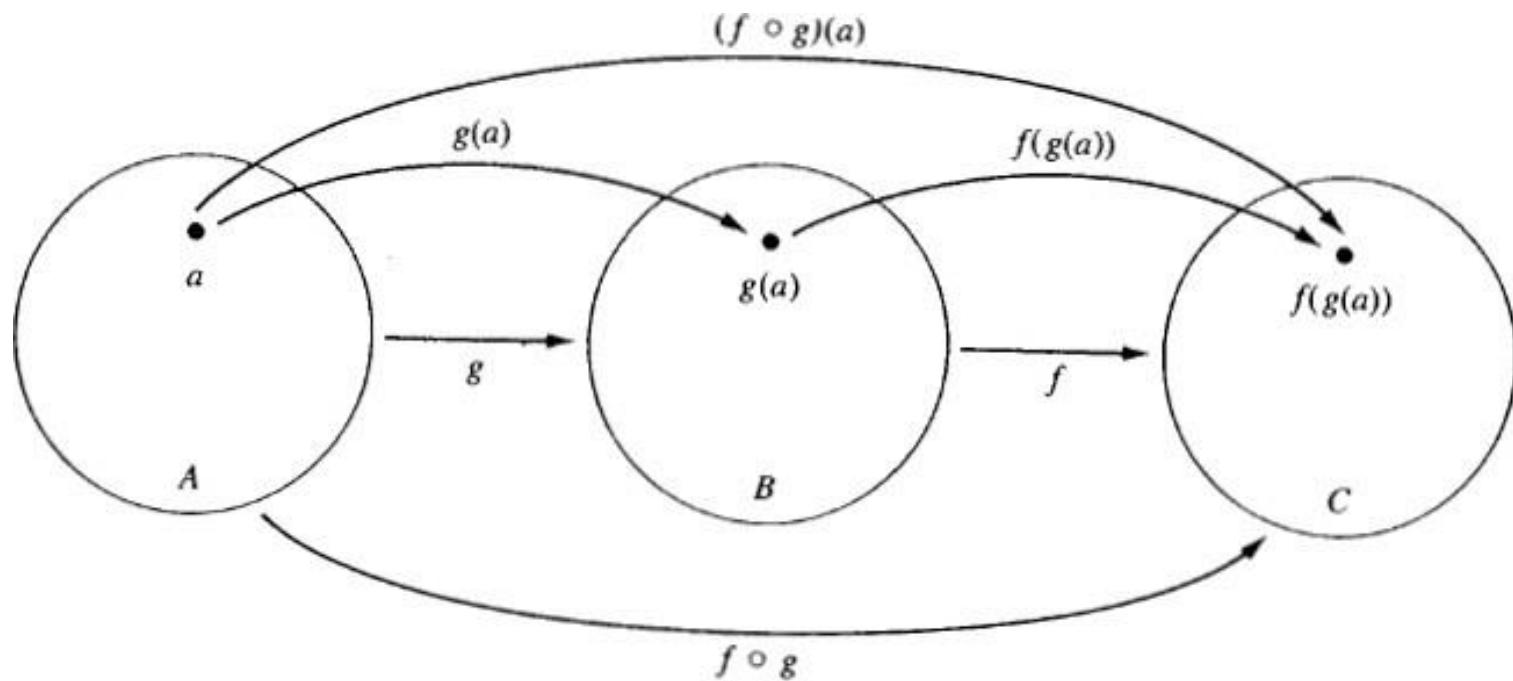
Function Composition Operator

Note the match here. It's necessary!

- For functions $g: A \rightarrow B$ and $f: B \rightarrow C$, there is a special operator called **compose** (“ \circ ”).
 - It composes (creates) a new function from f and g by applying f to the result of applying g .
 - We say $(f \circ g): A \rightarrow C$, where $(f \circ g)(a) = f(g(a))$.
 - Note: $f \circ g$ cannot be defined unless range of g is a subset of the domain of f .
 - Note $g(a) \in B$, so $f(g(a))$ is defined and $\in C$.
 - Note that \circ is non-commuting. (Like Cartesian \times , but unlike $+$, \wedge , \cup) (Generally, $f \circ g \neq g \circ f$.)

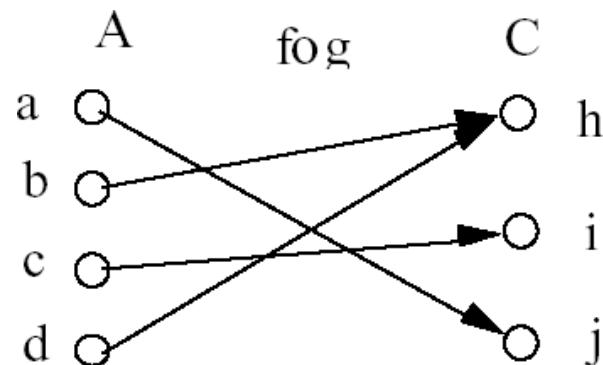
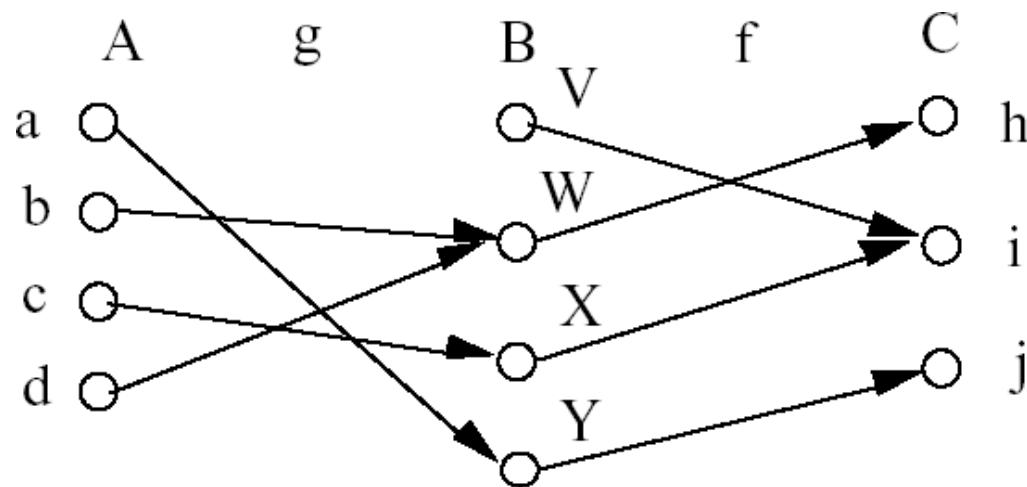
Function Composition Illustration

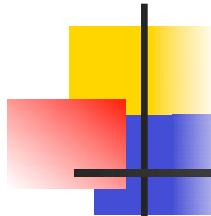
- $g: A \rightarrow B, f: B \rightarrow C$



Function Composition: Example

- $g: A \rightarrow B, f: B \rightarrow C$





Function Composition: Example

- Example 20: Let $g: \{a, b, c\} \rightarrow \{a, b, c\}$ such that $g(a) = b, g(b) = c, g(c) = a$.

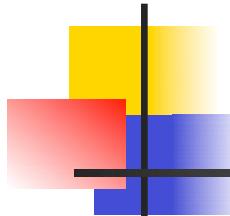
Let $f: \{a, b, c\} \rightarrow \{1, 2, 3\}$ such that
 $f(a) = 3, f(b) = 2, f(c) = 1$.

What is the composition of f and g , and what is the composition of g and f ?

- $f \circ g: \{a, b, c\} \rightarrow \{1, 2, 3\}$ such that
 $(f \circ g)(a) = 2, (f \circ g)(b) = 1, (f \circ g)(c) = 3$.

$$\begin{aligned}(f \circ g)(a) &= f(g(a)) = f(b) = 2 \\(f \circ g)(b) &= f(g(b)) = f(c) = 1 \\(f \circ g)(c) &= f(g(c)) = f(a) = 3\end{aligned}$$

- $g \circ f$ is not defined (why?)



Function Composition: Example

- If $f(x) = x^2$ and $g(x) = 2x + 1$, then what is the composition of f and g , and what is the composition of g and f ?
 - $$\begin{aligned}(f \circ g)(x) &= f(g(x)) \\ &= f(2x+1) \\ &= (2x+1)^2\end{aligned}$$
 - $$\begin{aligned}(g \circ f)(x) &= g(f(x)) \\ &= g(x^2) \\ &= 2x^2 + 1\end{aligned}$$

Note that $f \circ g \neq g \circ f$. ($4x^2+4x+1 \neq 2x^2+1$)

Images of Sets under Functions

- Given $f: A \rightarrow B$, and $S \subseteq A$,
- The ***image*** of S under f is simply the set of all images (under f) of the elements of S .

$$\begin{aligned}f(S) &= \{f(t) \mid t \in S\} \\&= \{b \mid \exists t \in S: f(t) = b\}.\end{aligned}$$

- Note the range of f can be defined as simply the image (under f) of f 's domain.

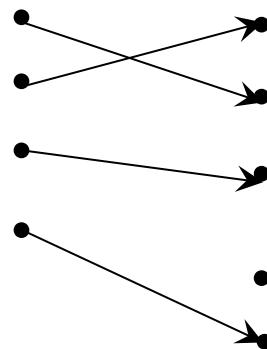
One-to-One Functions

- A function f is **one-to-one** (1–1), or **injective**, or an **injection**, iff $f(a) = f(b)$ implies that $a = b$ for all a and b in the domain of f (i.e. every element of its range has *only* 1 pre-image).
 - Formally, given $f : A \rightarrow B$,
“ f is injective”: $\forall a, b (f(a) = f(b) \rightarrow a = b)$ or
equivalently $\forall a, b (a \neq b \rightarrow f(a) \neq f(b))$
- Only one element of the domain is mapped to any given one element of the range.
 - Domain & range have the same cardinality.
What about codomain?

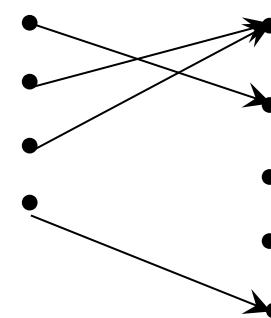


One-to-One Illustration

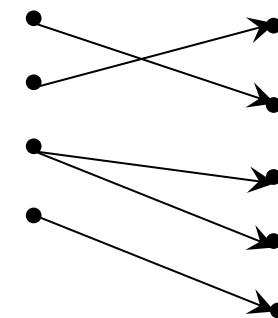
- Bipartite (2-part) graph representations of functions that are (or not) one-to-one:



One-to-one



Not one-to-one



Not even a function!

Example 8:

Is the function $f: \{a, b, c, d\} \rightarrow \{1, 2, 3, 4, 5\}$ with

$f(a) = 4$, $f(b) = 5$, $f(c) = 1$, and $f(d) = 3$ one-to-one?

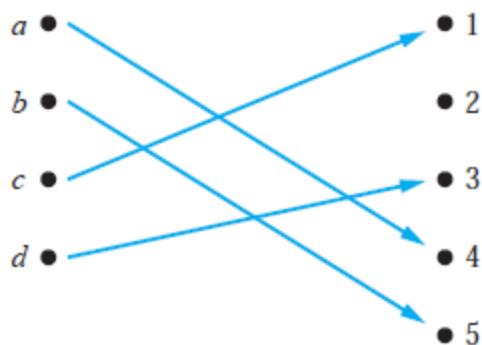


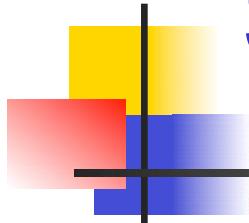
FIGURE 3 A One-to-One Function.

Example 9:

Let $f : \mathbf{Z} \rightarrow \mathbf{Z}$ such that $f(x) = x^2$. Is f one-to-one?

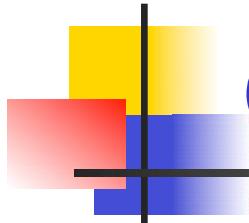
Solution: The function $f(x) = x^2$ is not one-to-one because, for instance, $f(1) = f(-1) = 1$, but $1 \neq -1$.

Note that the function $f(x) = x^2$ with its domain restricted to \mathbf{Z}^+ is one-to-one. (Technically, when we restrict the domain of a function, we obtain a new function whose values agree with those of the original function for the elements of the restricted domain. The restricted function is not defined for elements of the original domain outside of the restricted domain.)



Sufficient Conditions for 1–1ness

- For functions f over numbers, we say:
 - f is **strictly** (or **monotonically**) **increasing** iff $x > y \rightarrow f(x) > f(y)$ for all x, y in domain;
 - f is **strictly** (or **monotonically**) **decreasing** iff $x > y \rightarrow f(x) < f(y)$ for all x, y in domain;
- If f is either strictly increasing or strictly decreasing, then f is one-to-one.
 - E.g. x^3



Onto (Surjective) Functions

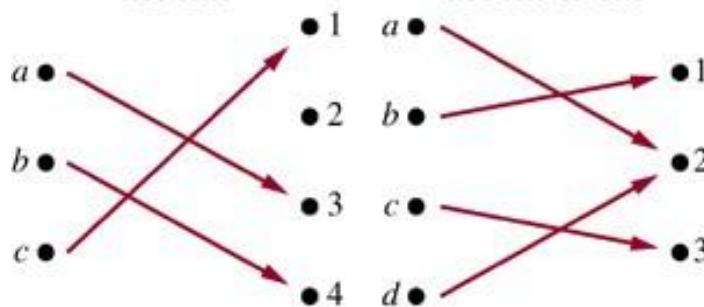
- A function $f: A \rightarrow B$ is **onto** or **surjective** or a **surjection** iff for every element $b \in B$ there is an element $a \in A$ with $f(a) = b$ ($\forall b \in B, \exists a \in A: f(a) = b$) (i.e. its range is equal to its codomain).
- Think: An *onto* function maps the set A onto (over, covering) the *entirety* of the set B , not just over a piece of it.
- *E.g.*, for domain & codomain \mathbb{R} , x^3 is onto, whereas x^2 isn't. (Why not?)

Illustration of Onto

- Some functions that are, or are not, *onto* their codomains:

© The McGraw-Hill Companies, Inc. all rights reserved.

(a) One-to-one,
not onto

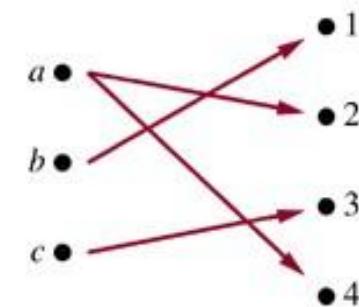
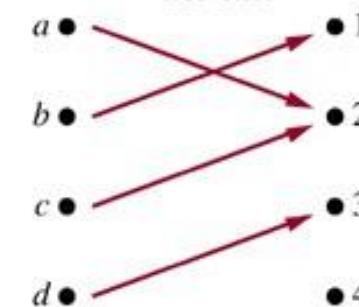
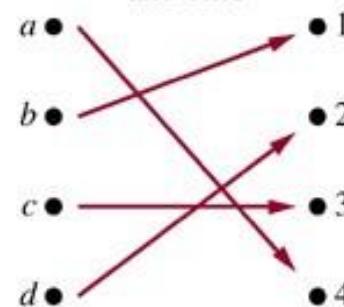


(b) Onto,
not one-to-one

(c) One-to-one,
and onto

(d) Neither one-to-one
nor onto

(e) Not a function



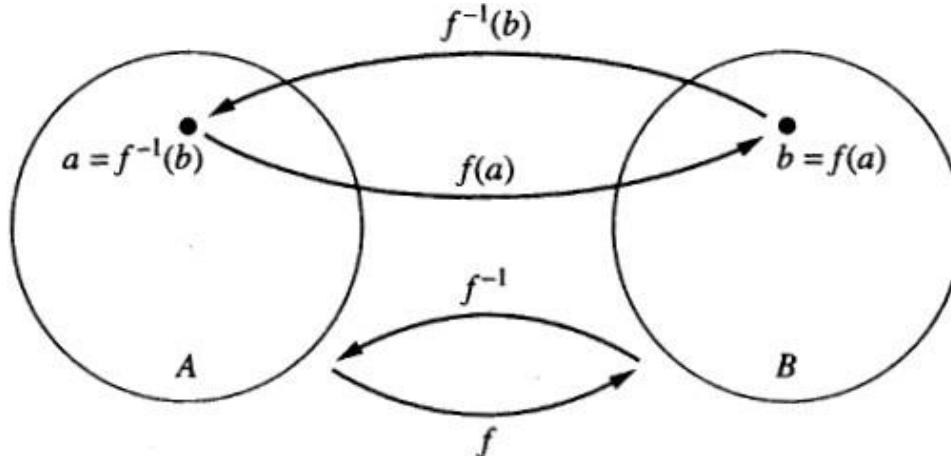
- Example 13: Is the function $f(x) = x + 1$ from the set of integers to the set of integers onto?

Bijections and Inverse Function

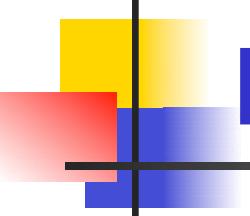
- A function f is said to be a ***one-to-one correspondence***, or a ***bijection***, or *reversible*, or *invertible*, iff it is both one-to-one and onto.
- Let $f: A \rightarrow B$ be a bijection.
The ***inverse function*** of f is the function that assigns to an element $b \in B$ the unique element $a \in A$ such that $f(a) = b$.
The inverse function of f is denoted by $f^{-1}: B \rightarrow A$.
Hence, $f^{-1}(b) = a$ when $f(a) = b$.

Inverse Function Illustration

- Let $f: A \rightarrow B$ be a bijection



- Example 16:** Let $f: \{a, b, c\} \rightarrow \{1, 2, 3\}$ such that $f(a) = 2$, $f(b) = 3$, $f(c) = 1$. Is f invertible, and if it is, what is its inverse? Yes. $f^{-1}(1) = c$, $f^{-1}(2) = a$, $f^{-1}(3) = b$
- Example 18:** Let f be the function from \mathbf{R} to \mathbf{R} with $f(x) = x^2$. Is f invertible? No. f is not a one-to-one function. So it's not invertible.

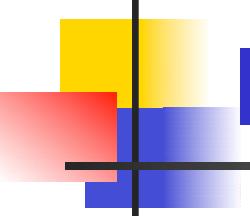


Mappings in Java

- A discrete function can be represented by a Map interface or HashMap class in Java programming language
 - ```
Map map<Integer, String>
 = new HashMap<Integer, String>();
```
  - Here, the domain is Integer, the codomain is String
- We can construct such a mapping by putting all pairs { $a$ ,  $f(a)$ } into our map. ( $a$  is the *key*,  $f(a)$  is the *value*.)
  - ```
map.put(2, "Jan");
```
 - ```
for (Kid kid:kids) {map.put(kid.id, kid.name);}
```
- If we put another pair with the same key, it will overwrite the previous pair – it's not a function! (May be a bug...)

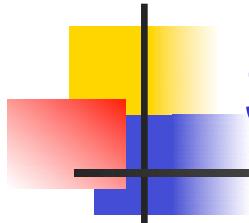
# Image, Range, Bijection in Java

- Map.keys () returns the **image**
  - it's a Java Set!
- map.values () returns the **range**
  - it's a Java Set!
- Is a map a bijection?  
Iff the cardinalities of the **image** and **range** are the same:
  - if (map.keys ().size () ==map.values ().size ()) {  
    System.out.println("map is a bijection");  
}



# Inverse Function in Java

- Let's construct an inverse!
- Prepare the inverse function:
  - Map inverse<String, Integer>  
= new HashMap<String, Integer>();
  - Here, the domain is String, the codomain is Integer
- Go through all keys in map (all elements of the **image**) and put each pair {value,key} into inverse:
  - ```
for (Integer id:map.keys()) {
    String name = map.get(id);
    inverse.put(id:name, id);
}
```



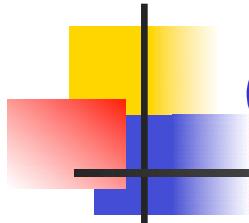
Summation Notation

- Given a sequence $\{a_n\}$, an integer *lower bound (or limit)* $j \geq 0$, and an integer *upper bound* $k \geq j$, then the *summation of $\{a_n\}$ from a_j to a_k* is written and defined as follows:

$$\sum_{i=j}^k a_i = a_j + a_{j+1} + \dots + a_k$$

- Here, *i* is called the *index of summation*.

$$\sum_{i=j}^k a_i = \sum_{m=j}^k a_m = \sum_{l=j}^k a_l$$



Generalized Summations

- For an infinite sequence, we write:

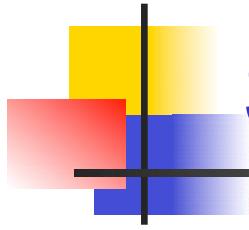
$$\sum_{i=j}^{\infty} a_i = a_j + a_{j+1} + \cdots$$

- To sum a function over all members of a set $X = \{x_1, x_2, \dots\}$:

$$\sum_{x \in X} f(x) = f(x_1) + f(x_2) + \cdots$$

- Or, if $X = \{x \mid P(x)\}$, we may just write:

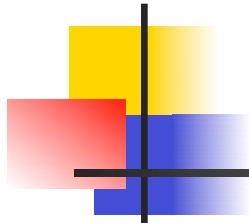
$$\sum_{P(x)} f(x) = f(x_1) + f(x_2) + \cdots$$



Simple Summation Example

- $\sum_{i=2}^4 (i^2 + 1) =$

- $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{100} = \sum_{i=1}^{100} \frac{1}{i}$



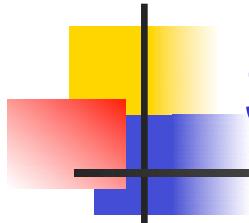
More Summation Examples

- An infinite sequence with a finite sum:

$$\sum_{i=0}^{\infty} 2^{-i} = 2^0 + 2^{-1} + \dots = 1 + \frac{1}{2} + \frac{1}{4} + \dots = 2$$

- Using a predicate to define a set of elements to sum over:

$$\sum_{\substack{(x \text{ is prime) } \wedge \\ x < 10}} x^2 = 2^2 + 3^2 + 5^2 + 7^2 \\ = 4 + 9 + 25 + 49 = 87$$



Summation Manipulations

- Some handy identities for summations:
 - Summing constant value

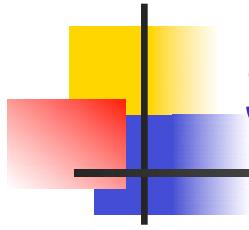
$$\sum_{n=i}^j c = (j - i + 1) \cdot c$$

Number of terms
in the summation

$$\sum_{n=1}^3 2 = \quad = 0$$

$$\sum_{n=-1}^2 2i$$

$$= 4 \oplus (2i) = 8i$$

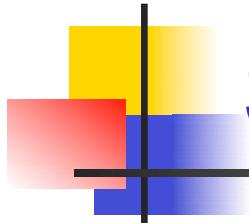


Summation Manipulations

- Distributive law

$$\sum_{n=i}^j cf(n) = c \sum_{n=i}^j f(n)$$

$$\begin{aligned}\sum_{n=1}^3 (4 \cdot n^2) &= 4 \cdot 1^2 + 4 \cdot 2^2 + 4 \cdot 3^2 \\&= 4 \cdot (1^2 + 2^2 + 3^2) \\&= 4 \sum_{n=1}^3 n^2\end{aligned}$$

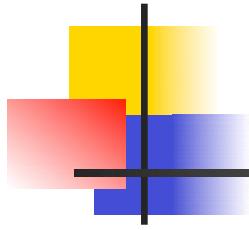


Summation Manipulations

- An application of commutativity

$$\sum_{n=i}^j (f(n) + g(n)) = \sum_{n=i}^j f(n) + \sum_{n=i}^j g(n)$$

$$\begin{aligned}\sum_{n=2}^4 (n + 2n) &= (2 + 2 \cdot 2) + (3 + 2 \cdot 3) + (4 + 2 \cdot 4) \\&= (2 + 3 + 4) + (2 \cdot 2 + 2 \cdot 3 + 2 \\&\quad \cdot 4) \\&= \sum_{n=2}^4 n + \sum_{n=2}^4 2n\end{aligned}$$



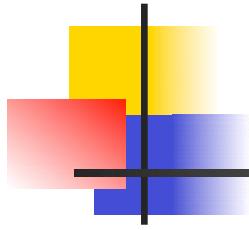
Index Shifting

$$\sum_{i=j}^m f(i) = \sum_{k=j+n}^{m+n} f(k-n)$$

$$\sum_{i=1}^4 i^2 = 1^2 + 2^2 + 3^2 + 4^2$$

- Let $k = i + 2$, then $i = k - 2$

$$\begin{aligned}\sum_{k=1+2}^{4+2} (k-2)^2 &= \sum_{k=3}^6 (k-2)^2 \\ &= (3-2)^2 + (4-2)^2 + (5-2)^2 + (6-2)^2\end{aligned}$$

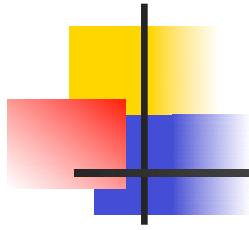


More Summation Manipulations

- Sequence splitting

$$\sum_{i=j}^k f(i) = \sum_{i=j}^m f(i) + \sum_{i=m+1}^k f(i) \quad \text{if } j \leq m < k$$

$$\begin{aligned}\sum_{i=0}^4 i^3 &= 0^3 + 1^3 + 2^3 + 3^3 + 4^3 \\&= (0^3 + 1^3 + 2^3) + (3^3 + 4^3) \\&= \sum_{i=0}^2 i^3 + \sum_{i=3}^4 i^3\end{aligned}$$

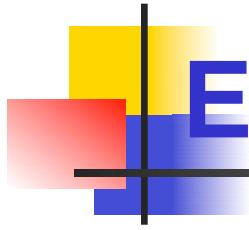


More Summation Manipulations

- Order reversal

$$\sum_{i=0}^k f(i) = \sum_{i=0}^k f(k-i)$$

$$\begin{aligned}\sum_{i=0}^3 i^3 &= 0^3 + 1^3 + 2^3 + 3^3 \\&= (3-0)^3 + (3-1)^3 + (3-2)^3 + (3-3)^3 \\&= \sum_{i=0}^3 (3-i)^3\end{aligned}$$

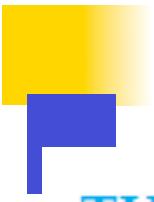


Example: Geometric Progression

- A geometric progression is a sequence of the form $a, ar, ar^2, ar^3, \dots, ar^m, \dots$ where $a, r \in \mathbb{R}$.
- The sum of such a sequence is given by:

$$S = \sum_{i=0}^n ar^i$$

- We can reduce this to *closed form* via clever manipulation of summations...



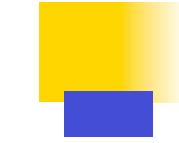
THEOREM 1

If a and r are real numbers and $r \neq 0$, then

$$\sum_{j=0}^n ar^j = \begin{cases} \frac{ar^{n+1} - a}{r - 1} & \text{if } r \neq 1 \\ (n + 1)a & \text{if } r = 1. \end{cases}$$

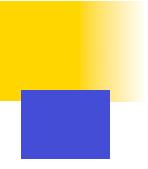
Proof: Let

$$S_n = \sum_{j=0}^n ar^j.$$



To compute S , first multiply both sides of the equality by r and then manipulate the resulting sum as follows:

$$\begin{aligned} rS_n &= r \sum_{j=0}^n ar^j && \text{substituting summation formula for } S \\ &= \sum_{j=0}^n ar^{j+1} && \text{by the distributive property} \\ &= \sum_{k=1}^{n+1} ar^k && \text{shifting the index of summation, with } k = j + 1 \\ &= \left(\sum_{k=0}^n ar^k \right) + (ar^{n+1} - a) && \text{removing } k = n + 1 \text{ term and adding } k = 0 \text{ term} \\ &= S_n + (ar^{n+1} - a) && \text{substituting } S \text{ for summation formula} \end{aligned}$$



From these equalities, we see that

$$r S_n = S_n + (ar^{n+1} - a).$$

Solving for S_n shows that if $r \neq 1$, then

$$S_n = \frac{ar^{n+1} - a}{r - 1}.$$

If $r = 1$, then the $S_n = \sum_{j=0}^n ar^j = \sum_{j=0}^n a = (n + 1)a$.

Gauss' Trick, Illustrated

- Consider the sum:

$$1+2+\dots+(n/2)+((n/2)+1)+\dots+(n-1)+n$$

The diagram shows the sum $1+2+\dots+(n/2)+((n/2)+1)+\dots+(n-1)+n$. It uses red ovals to group pairs of terms: $(1, n)$, $(2, n-1)$, ..., $((n/2), (n/2)+1)$. A red curve connects the first term (1) to the last term (n). The number of pairs is labeled $n+1$. The middle term is labeled $n+1$. Ellipses indicate intermediate terms between the pairs.

- We have $n/2$ pairs of elements, each pair summing to $n+1$, for a total of $(n/2)(n+1)$.

Some Shortcut Expressions

TABLE 2 Some Useful Summation Formulae.

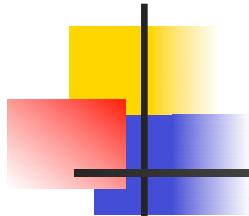
<i>Sum</i>	<i>Closed Form</i>
$\sum_{k=0}^n ar^k \ (r \neq 0)$	$\frac{ar^{n+1} - a}{r - 1}, r \neq 1$
$\sum_{k=1}^n k$	$\frac{n(n + 1)}{2}$
$\sum_{k=1}^n k^2$	$\frac{n(n + 1)(2n + 1)}{6}$
$\sum_{k=1}^n k^3$	$\frac{n^2(n + 1)^2}{4}$
$\sum_{k=0}^{\infty} x^k, x < 1$	$\frac{1}{1 - x}$
$\sum_{k=1}^{\infty}, kx^{k-1}, x < 1$	$\frac{1}{(1 - x)^2}$

Geometric sequence

Gauss' trick

Quadratic series

Cubic series



Using the Shortcuts

- Example: Evaluate

$$\sum_{k=50}^{100} k^2$$

- Use series splitting.

■ Solve
for desired
summation.

■ Apply
quadratic series
rule.

$$\begin{aligned}\sum_{k=1}^{100} k^2 &= \left(\sum_{k=1}^{49} k^2 \right) + \sum_{k=50}^{100} k^2 \\&= \left(\sum_{k=1}^{100} k^2 \right) - \sum_{k=1}^{49} k^2 \\&= \frac{100 \cdot 101 \cdot 201}{6} - \frac{49 \cdot 50 \cdot 99}{6} \\&= 338,350 - 40,425 \\&= 297,925.\end{aligned}$$

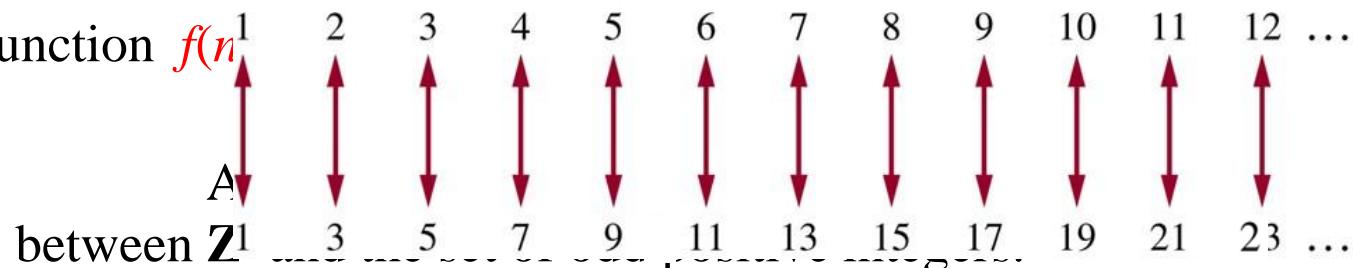
- Evaluate.

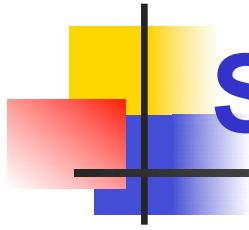
Cardinality

- The sets A and B have the same **cardinality** if and only if there is a one-to-one correspondence from A to B .
- A set that is either finite or has the same cardinality as the set of positive integers is called **countable**.
- A set that is not countable is called **uncountable**.
- Example: Show that the set of odd positive integers is a countable set.

© The McGraw-Hill Companies, Inc. all rights reserved.

Consider the function $f(n)$
integers





Summation Manipulations

- Useful identities:

$$\sum_{i=j}^k f(i) = \sum_{i=j}^m f(i) + \sum_{i=m+1}^k f(i) \quad \text{if } j \leq m < k$$

(Sequence splitting.)

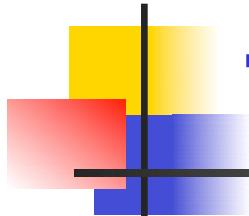
$$\sum_{i=0}^k f(i) = \sum_{i=0}^k f(k-i) \quad \text{(Order reversal.)}$$

$$\sum_{i=1}^{2k} f(i) = \sum_{i=1}^k (f(2i-1) + f(2i)) \quad \text{(Grouping.)}$$



Chapter 3. The Fundamentals

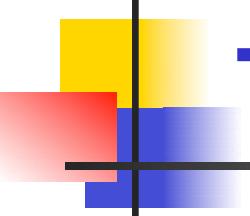
- 4. The Integers and Division
- 5. Primes and Greatest Common Divisors



The Division “Algorithm”

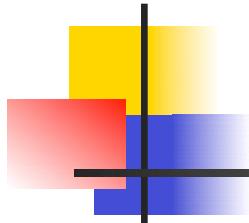
- It's really just a *theorem*, not an algorithm...
- Only called an “algorithm” for historical reasons.
- **Theorem:** For any integer ***dividend* a** and ***divisor* $d \in \mathbb{Z}^+$** , there are unique integers ***quotient* q** and ***remainder* $r \in \mathbb{N}$** such that **$a = dq + r$** and **$0 \leq r < d$** .

$$q = a \text{ } div \text{ } d, r = a \text{ } mod \text{ } d.$$



The mod Operator

- An integer “division remainder” operator.
- Let $a, d \in \mathbb{Z}$ with $d > 1$. Then $a \bmod d$ denotes the remainder r from the division “algorithm” with dividend a and divisor d ; i.e. the remainder when a is divided by d .
Also, $a \text{ div } d$ denotes the quotient q .
- We can compute $(a \bmod d)$ by:
$$a - d \cdot \lfloor a/d \rfloor$$
- In C/C++/Java languages, “%” = mod.

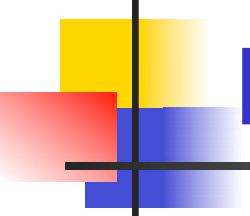


The mod Operator: Examples

- $101 = 11 \cdot 9 + 2$ (dividend: 101, divisor: 11)
 - $101 \text{ div } 11 = 9$ $101 \text{ mod } 11 = 2$

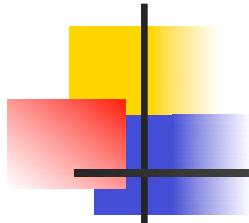
- $-11 = 3 \cdot (-4) + 1$ or ~~$-11 = 3 \cdot (-3) - 2 ?$~~
(dividend: -11, divisor: 3)
 - $-11 \text{ div } 3 = -4$ $-11 \text{ mod } 3 = 1$
(quotient: -4, remainder: 1)

- Note that the remainder must not be negative.



Modular Congruence

- Let $a, b \in \mathbb{Z}$, $m \in \mathbb{Z}^+$, where $\mathbb{Z}^+ = \{n \in \mathbb{Z} \mid n > 0\} = \mathbb{N} - \{0\}$ (the positive integers).
- Then a is congruent to b modulo m , written “ $a \equiv b \pmod{m}$ ”, iff $m|(a - b)$.
- Note: this is a different use of “ \equiv ” than the meaning “equivalent” or “is defined as” used before.
- It’s also equivalent to: $(a - b) \bmod m = 0$.
- E.g. $17 \equiv 5 \pmod{6}$, $24 \not\equiv 14 \pmod{6}$



Useful Congruence Theorems

■ **Theorem:** Let $a, b \in \mathbb{Z}$, $m \in \mathbb{Z}^+$. Then:

$$a \equiv b \pmod{m} \Leftrightarrow a \bmod m = b \bmod m.$$

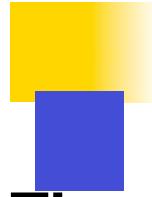
■ **Theorem:** Let $a, b \in \mathbb{Z}$, $m \in \mathbb{Z}^+$. Then:

$$a \equiv b \pmod{m} \Leftrightarrow \exists k \in \mathbb{Z}: a = b + km.$$

Proof: If $a \equiv b \pmod{m}$, by the definition of congruence we know that $m \mid (a - b)$. This means that there is an integer k such that $a - b = km$, so that $a = b + km$.

Conversely, if there is an integer k such that $a = b + km$, then $km = a - b$. Hence, m divides $a - b$, so that $a \equiv b \pmod{m}$.

■ **Theorem:** Let $a, b, c, d \in \mathbb{Z}$, $m \in \mathbb{Z}^+$. Then if $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, then:



Theorem: Let $a, b, c, d \in \mathbb{Z}$, $m \in \mathbb{Z}^+$. Then if

$a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, then:

- $a + c \equiv b + d \pmod{m}$, and
- $ac \equiv bd \pmod{m}$

Proof: We use a direct proof. Because $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, by Theorem 4 there are integers s and t with $b = a + sm$ and $d = c + tm$. Hence,

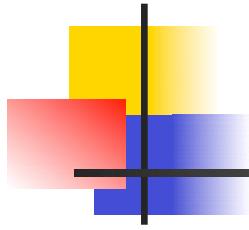
$$b + d = (a + sm) + (c + tm) = (a + c) + m(s + t)$$

and

$$bd = (a + sm)(c + tm) = ac + m(at + cs + stm).$$

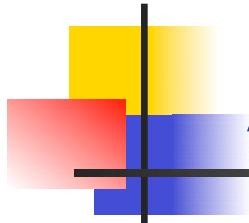
Hence,

$$a + c \equiv b + d \pmod{m} \text{ and } ac \equiv bd \pmod{m}.$$



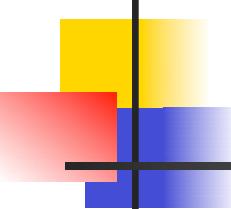
Congruence Theorem Example

- $7 \equiv 2 \pmod{5}$ and $11 \equiv 1 \pmod{5}$.
 - $7 + 11 = 18$ and $2 + 1 = 3$
Therefore, $7 + 11 \equiv 2 + 1 \pmod{5}$
 - $7 \times 11 = 77$ and $2 \times 1 = 2$
Therefore, $7 \times 11 \equiv 2 \times 1 \pmod{5}$



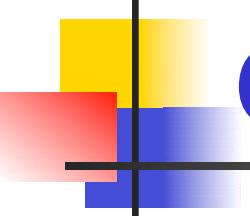
Applications of Congruence

- Hashing Functions (hashes)
- Pseudorandom Numbers
- Cryptology
- Universal Product Codes
- International Standard Book Numbers



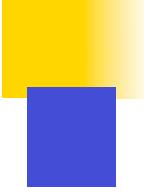
Hashing Functions

- We want to quickly store and retrieve records in memory locations.
- A hashing function takes a data item to be stored or retrieved and computes the first choice for a location for the item.
- $h(k) = k \bmod m$
- A hashing function h assigns memory location $h(k)$ to the record that has k as its key.
- $h(064212848) = 064212848 \bmod 111 = 14$
- $h(037149212) = 037149212 \bmod 111 = 65$
- $h(107405723) = 107405723 \bmod 111 = 14 \Rightarrow \text{collision!}$
- Find the first unoccupied memory location after the occupied memory.
- In this case, assign memory location 15.
- If collision occurs infrequently, and if when one does occur it is resolved quickly, then hashing provides a very fast method of storing and retrieving data.



Cryptology (I)

- The study of secret messages
- **Encryption** is the process of making a message secret. **Decryption** is the process of determining the original message from the encrypted message.
- Some simple early codes include *Caesar's cipher*:
 - Assign an integer from 0 to 25 to each letter based on its position in the alphabet.
 - Caesar's encryption method: $f(p) = (p + 3) \bmod 26$
 - Caesar's decryption method: $f^{-1}(p) = (p - 3) \bmod 26$
 - MEET YOU IN THE PARK \Rightarrow
PHHW BRX LQ WKH SDUN



EXAMPLE

What is the secret message produced from the message “MEET YOU IN THE PARK” using the Caesar cipher?

Solution: First replace the letters in the message with numbers.

This produces

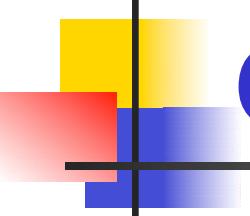
12 4 4 19 24 14 20 8 13 19 7 4 15 0 17 10.

Now replace each of these numbers p by $f(p) = (p + 3) \bmod 26$.

This gives

15 7 7 22 1 17 23 11 16 22 10 7 18 3 20 13.

Translating this back to letters produces the encrypted message “PHHW BRX LQ WKH SDUN.”



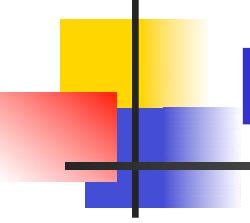
Cryptology (II)

- Caesar's encryption method does not provide a high level of security
- A slightly better approach: $f(p) = (ap + b) \bmod 26$

- **Example 10:**

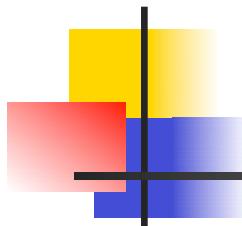
What letter replaces the letter K when the function $f(p) = (7p + 3) \bmod 26$ is used for encryption?

- 10 represents K
- $f(10) = (7 \times 10 + 3) \bmod 26 = 73 \bmod 26 = 21$
- 21 represents V
- Therefore, K is replaced by V in the encrypted message



Prime Numbers

- An integer $p > 1$ is **prime** iff the only positive factors of p are 1 and p itself.
- Some primes: 2, 3, 5, 7, 11, 13,...
- Non-prime integers greater than 1 are called **composite**, because they can be composed by multiplying two integers greater than 1.



The Fundamental Theorem of Arithmetic

Its "Prime Factorization"

- Every positive integer greater than 1 has a *unique* representation as a prime or as the product of a non-decreasing series of two or more primes.

Some examples:

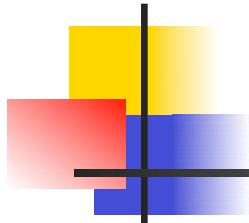
■ 2 = 2 (a prime 2)

■ 4 = 2 · 2 = 2²(product of series 2, 2)

$$\blacksquare 2000 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 5 \cdot 5 \cdot 5 = 2^4 \cdot 5^3 \quad 2001 = 3 \cdot 23 \cdot 29$$

$$2002 = 2 \cdot 7 \cdot 11 \cdot 13$$

2003 = 2003 (no clear pattern!)



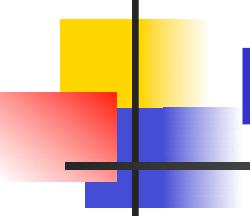
Prime Numbers: Theorems

- Contrapositive of Theorem 2:

An integer is prime if it is not divisible by any prime less than or equal to its square root \sqrt{n}

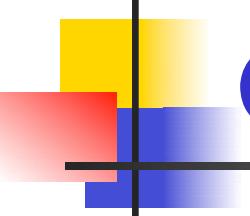
- Example: Show that 101 is prime

- Primes not exceeding $\sqrt{101}$: 2, 3, 5, 7
- 101 is not divisible by any of 2, 3, 5, or 7
- Therefore, 101 is a prime



Prime Factorization

- **Example 4:** Find the prime factorization of 7007 ($\sqrt{7007} \approx 83.7$)
 - Perform division of 7007 by successive primes
 $7007 / 7 = 1001$ ($7007 = 7 \cdot 1001$)
 - Perform division of 1001 by successive primes beginning with 7
 $1001 / 7 = 143$ ($7007 = 7 \cdot 7 \cdot 143$)
 - Perform division of 143 by successive primes beginning with 7
 $143 / 11 = 13$ ($7007 = 7 \cdot 7 \cdot 11 \cdot 13$
 $= 7^2 \cdot 11 \cdot 13$)



Greatest Common Divisor

- The ***greatest common divisor*** $\gcd(a,b)$ of integers a, b is the largest integer d that is a divisor both of a and of b .

$$d = \gcd(a,b) = \max(d: d|a \wedge d|b)$$

$$\Leftrightarrow d|a \wedge d|b \wedge \forall e \in \mathbb{Z}, (e|a \wedge e|b) \rightarrow (d \geq e)$$

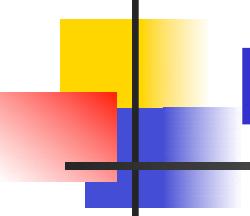
- **Example:** $\gcd(24,36) = ?$

- Positive divisors of 24: 1, 2, 3, 4, 6, 8, 12, 24

- Positive divisors of 36: 1, 2, 3, 4, 6, 9, 12, 18, 36

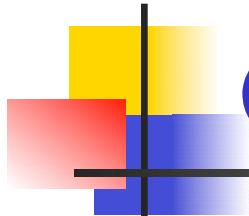
- Positive common divisors: 1, 2, 3, 4, 6, 12.

The largest one of these is 12.



Relative Primality

- Integers a and b are called ***relatively prime*** or ***coprime*** iff their $\gcd = 1$.
- **Example:** Neither 21 nor 10 is prime, but they are *relatively prime*. (divisors of 21: **1, 3, 7, 21**; divisors of 10: **1, 2, 5, 10**; so they have no common factors > 1 , so their $\gcd = 1$.)
- A set of integers $\{a_1, a_2, a_3, \dots\}$ is ***pairwise relatively prime*** if all pairs (a_i, a_j) , for $i \neq j$, are relatively prime.



GCD Shortcut

- If the prime factorizations are written as

$$a = p_1^{a_1} p_2^{a_2} \cdots p_n^{a_n} \text{ and } b = p_1^{b_1} p_2^{b_2} \cdots p_n^{b_n},$$

then the GCD is given by:

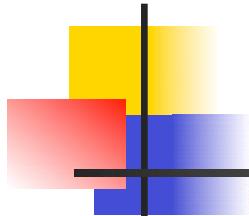
$$\gcd(a, b) = p_1^{\min(a_1, b_1)} p_2^{\min(a_2, b_2)} \cdots p_n^{\min(a_n, b_n)}.$$

- Example of using the shortcut:

- $a = 84 = \underline{2} \cdot \underline{2} \cdot \underline{3} \cdot 7 = 2^2 \cdot 3^1 \cdot 7^1$

- $b = 96 = \underline{2} \cdot \underline{2} \cdot 2 \cdot 2 \cdot \underline{2} \cdot \underline{3} = 2^5 \cdot 3^1 \cdot 7^0$

- $\gcd(84, 96) = 2^2 \cdot 3^1 \cdot 7^0 = 2 \cdot 2 \cdot 3 = 12.$



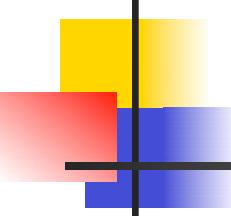
Least Common Multiple

- $\text{lcm}(a,b)$ of positive integers a, b , is the smallest positive integer that is a multiple both of a and of b . *E.g.* $\text{lcm}(6,10) = 30$

$$m = \text{lcm}(a,b) = \min(m: a|m \wedge b|m)$$

$$\Leftrightarrow a|m \wedge b|m \wedge \forall n \in \mathbb{Z}: (a|n \wedge b|n) \rightarrow (m \leq n)$$

- **Example:** $\text{lcm}(24,36) = ?$
 - Positive multiples of 24: 24, 48, 72, 96, 120, 144, ...
 - Positive multiples of 36: 36, 72, 108, 144, ...
 - Positive common multiples: 72, 144, ... The smallest one of these is 72.



LCM Shortcut

- If the prime factorizations are written as

$$a = p_1^{a_1} p_2^{a_2} \dots p_n^{a_n} \text{ and } b = p_1^{b_1} p_2^{b_2} \dots p_n^{b_n},$$

then the LCM is given by

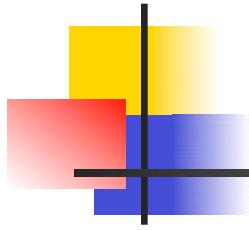
$$\text{lcm}(a, b) = p_1^{\max(a_1, b_1)} p_2^{\max(a_2, b_2)} \dots p_n^{\max(a_n, b_n)}.$$

- Example of using the shortcut:

- $a = 84 = 2 \cdot 2 \cdot 3 \cdot 7 = 2^2 \cdot 3^1 \cdot 7^1$

- $b = 96 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 3 = 2^5 \cdot 3^1 \cdot 7^0$

- $\text{lcm}(84, 96) = 2^5 \cdot 3^1 \cdot 7^1 = 32 \cdot 3 \cdot 7 = 672$



LCM: Another Example

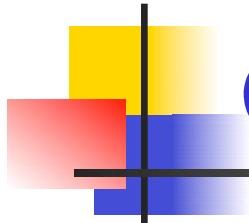
■■ Example 15:

What is the least common multiple of
 $2^3 \cdot 3^5 \cdot 7^2$ and $2^4 \cdot 3^3$?

■■ Solution: $\text{lcm}(2^3 \cdot 3^5 \cdot 7^2, 2^4 \cdot 3^3)$

$$= 2^{\max(3,4)} \cdot 3^{\max(5,3)} \cdot 7^{\max(2,0)}$$

$$= 2^4 \cdot 3^5 \cdot 7^2$$



GCD and LCM

■■ **Theorem:** Let a and b be positive integers. Then

$$ab = \gcd(a,b) \times \text{lcm}(a,b)$$

■■ **Example**

$$\blacksquare a = 84 = 2 \cdot 2 \cdot 3 \cdot 7 = 2^2 \cdot 3^1 \cdot 7^1$$

$$\blacksquare b = 96 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 3 = 2^5 \cdot 3^1 \cdot 7^0$$

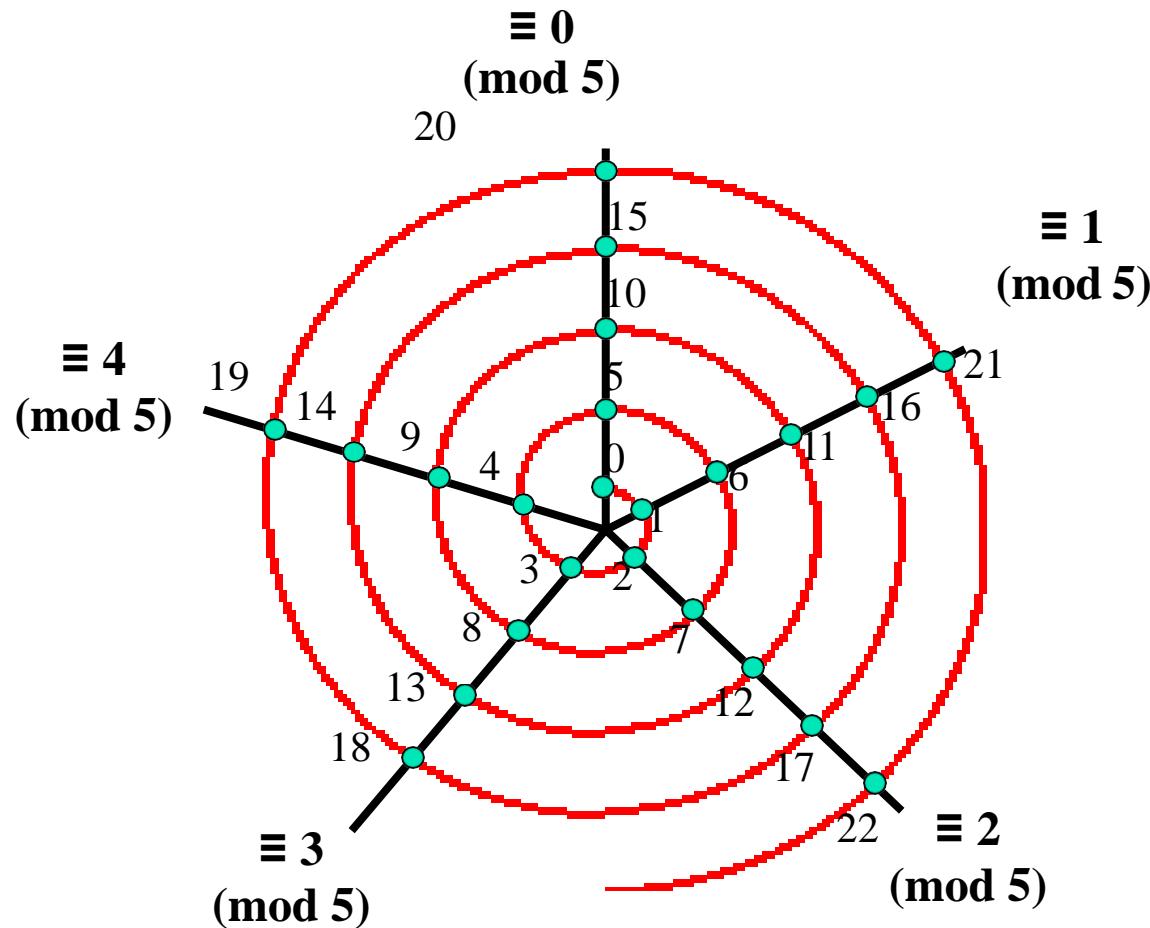
$$\blacksquare ab = (2^2 \cdot 3^1 \cdot 7^1) \cdot (2^5 \cdot 3^1 \cdot 7^0) = 2^2 \cdot 3^1 \cdot 7^0 \cdot 2^5 \cdot 3^1 \cdot 7^1$$

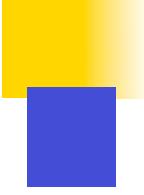
$$= 2^{\min(2,5)} \cdot 3^{\min(1,1)} \cdot 7^{\min(1,0)} \cdot 2^{\max(2,5)} \cdot 3^{\max(1,1)} \cdot 7^{\max(1,0)}$$

$$= \gcd(a,b) \times \text{lcm}(a,b)$$

Spiral Visualization of mod

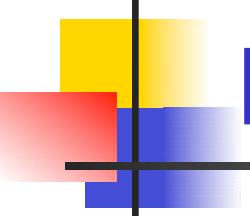
- ## ■ Example shown: modulo-5 arithmetic





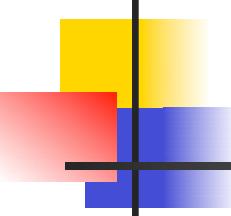
Pseudorandom Numbers

- Randomly chosen numbers are often needed for computer simulations.
- Different methods have been devised for generating numbers that have properties of randomly chosen numbers.
- Because numbers generated by systematic methods are not truly random, they are called **pseudorandom numbers**.
- The most commonly used procedure for generating pseudorandom numbers is the **linear congruential method**.



Linear Congruential Method

- Requires four natural numbers:
 - The *modulus m*, the *multiplier a*, the *increment c*, and the seed x_0 .
 - where $2 \leq a < m$, $0 \leq c < m$, $0 \leq x_0 < m$.
- Generates the pseudo-random sequence $\{x_n\}$ with $0 \leq x_n < m$, via the following:
$$x_{n+1} = (ax_n + c) \bmod m$$
- Tends to work best when a , c , m are prime, or at least relatively prime.
- If $c = 0$, the method is called a *pure multiplicative generator*.



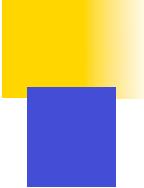
Example

- Find the sequence of pseudorandom numbers generated by the linear congruential method with modulus $m = 9$, multiplier $a = 7$, increment $c = 4$, and seed $x_0 = 3$.
Solution: We compute the terms of this sequence by successively using the recursively defined function $x_{n+1} = (7x_n + 4) \bmod 9$, beginning by inserting the seed $x_0 = 3$ to find x_1 . We find that

$$x_1 = 7x_0 + 4 \bmod 9 = 7 \cdot 3 + 4 \bmod 9 = 25 \bmod 9 = 7,$$

$$x_2 = 7x_1 + 4 \bmod 9 = 7 \cdot 7 + 4 \bmod 9 = 53 \bmod 9 = 8,$$

$$x_3 = 7x_2 + 4 \bmod 9 = 7 \cdot 8 + 4 \bmod 9 = 60 \bmod 9 = 6,$$


$$\begin{aligned}x_4 &= 7 \times 3 + 4 \bmod 9 = 7 \quad \cdot \quad 6 + 4 \bmod 9 = 46 \bmod 9 = 1, \\x_5 &= 7 \times 4 + 4 \bmod 9 = 7 \quad \cdot \quad 1 + 4 \bmod 9 = 11 \bmod 9 = 2, \\x_6 &= 7 \times 5 + 4 \bmod 9 = 7 \quad \cdot \quad 2 + 4 \bmod 9 = 18 \bmod 9 = 0, \\x_7 &= 7 \times 6 + 4 \bmod 9 = 7 \quad \cdot \quad 0 + 4 \bmod 9 = 4 \bmod 9 = 4, \\x_8 &= 7 \times 7 + 4 \bmod 9 = 7 \quad \cdot \quad 4 + 4 \bmod 9 = 32 \bmod 9 = 5, \\x_9 &= 7 \times 8 + 4 \bmod 9 = 7 \quad \cdot \quad 5 + 4 \bmod 9 = 39 \bmod 9 = 3.\end{aligned}$$

Because $x_9 = x_0$ and because each term depends only on the previous term, we see that the sequence

$$3, 7, 8, 6, 1, 2, 0, 4, 5, 3, 7, 8, 6, 1, 2, 0, 4, 5, 3, \dots$$

is generated. This sequence contains nine different numbers before repeating.

Mersenne Primes

■ ■ **Definition:** A *Mersenne prime* is a prime number of the form $2^p - 1$, where p is prime.

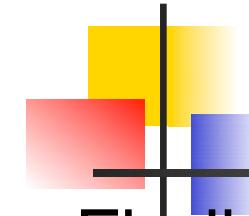
prime p	$2^p - 1$	Mersenne?
2	$2^2 - 1 = 3$	yes
3	$2^3 - 1 = 7$	yes
5	$2^5 - 1 = 31$	yes
7	$2^7 - 1 = 127$	yes
11,213	$2^{11,213} - 1$	yes
19,937	$2^{19,937} - 1$	yes
3,021,377	$2^{3,021,377} - 1$	Yes (late 1998)
43,112,609	$2^{43,112,609} - 1$	Yes (MID 2008)

largest Mersenne prime known
(with almost 13 million digits)



Chapter 3. The Fundamentals

3.6 Applications of Integers Algorithms



Euclid's Algorithm for GCD

- Finding GCDs by comparing prime factorizations can be difficult when the prime factors are not known!
- More efficient method of finding gcd is the called the **Euclidean algorithm**
- **Euclid discovered:** Let $a = bq + r$, where a, b, q , and r are integers. Then $\gcd(a,b) = \gcd(b,r)$ (i.e. $\gcd(a,b) = \gcd(b, (a \bmod b))$)
 - Example: $\gcd(36, 24) = \gcd(24, 12)$
- Sort a, b so that $a > b$, and then (given $b > 1$) $(a \bmod b) < b$, so problem is simplified.

The Euclidean Algorithm

- Example:

$$\gcd(91, 287).$$

First, divide 287, the larger of the two integers, by 91, the smaller, to obtain $287 = 91 \cdot 3 + 14$.

Any divisor of 91 and 287 must also be a divisor of $287 - 91 \cdot 3 = 14$.

any divisor of 91 and 14 must also be a divisor of $287 = 91 \cdot 3 + 14$.

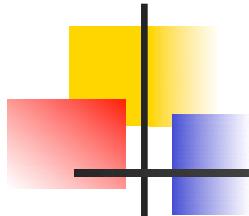
Hence, the $\gcd(91, 287)$ is the same as the $\gcd(91, 14)$ i.e. finding $\gcd(91, 287)$ reduced to find $\gcd(91, 14)$.

Next, divide 91 by 14 to obtain $91 = 14 \cdot 6 + 7$.

any common divisor of 91 and 14 also divides $91 - 14 \cdot 6 = 7$ and any common divisor of 14 and 7 divides 91,

$$\gcd(91, 14) = \gcd(14, 7). \quad \gcd(14, 7) = 7.$$

$$\gcd(287, 91) = \gcd(91, 14) = \gcd(14, 7) = 7$$



Euclid's Algorithm Example

- $\gcd(372, 164) = \gcd(164, 372 \text{ mod } 164)$
 - $372 \text{ mod } 164 = 372 - 164 \lfloor \frac{372}{164} \rfloor$
 $= 372 - 164 \cdot 2 = 372 - 328 = 44$
- $\gcd(164, 44) = \gcd(44, 164 \text{ mod } 44)$
 - $164 \text{ mod } 44 = 164 - 44 \lfloor \frac{164}{44} \rfloor$
 $= 164 - 44 \cdot 3 = 164 - 132 = 32$
- $\gcd(44, 32) = \gcd(32, 44 \text{ mod } 32) = \gcd(32, 12)$
 $= \gcd(12, 32 \text{ mod } 12) = \gcd(12, 8)$
 $= \gcd(8, 12 \text{ mod } 8) = \gcd(8, 4)$
 $= \gcd(4, 8 \text{ mod } 4) = \gcd(4, 0) = 4$



Find the greatest common divisor of 414 and 662 using the Euclidean algorithm.

Solution: Successive uses of the division algorithm give:

$$662 = 414 \cdot 1 + 248$$

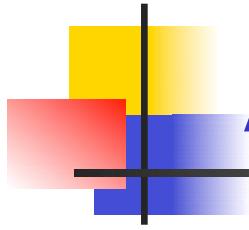
$$414 = 248 \cdot 1 + 166$$

$$248 = 166 \cdot 1 + 82$$

$$166 = 82 \cdot 2 + 2$$

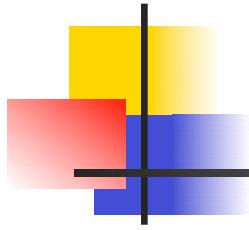
$$82 = 2 \cdot 41.$$

Hence, $\gcd(414, 662) = 2$, because 2 is the last nonzero remainder.



Applications

- Linear congruences
- Chinese Remainder Theorem
- Public Key Cryptography
 - The Rivest-Shamir-Adleman (RSA) cryptosystem



■ Theorem 1:

- $\forall a,b \in \mathbf{Z}^+$: $\exists s,t \in \mathbf{Z}$: $\gcd(a,b) = sa + tb$

■ Lemma 1:

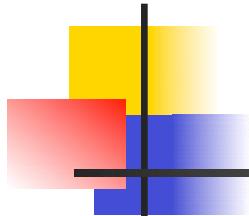
- $\forall a,b,c \in \mathbf{Z}^+$: $\gcd(a,b)=1 \wedge a \mid bc \rightarrow a \mid c$

■ Lemma 2:

- If p is prime and $p \mid a_1 a_2 \dots a_n$ (integers a_i)
then $\exists i$: $p \mid a_i$.

■ Theorem 2:

- If $ac \equiv bc \pmod{m}$ and $\gcd(c,m)=1$, then
 $a \equiv b \pmod{m}$. ($m \in \mathbf{Z}^+$, $a,b,c \in \mathbf{Z}$)



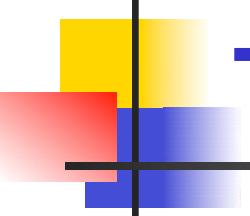
BÉZOUT'S THEOREM

- $\forall a, b \in \mathbb{Z}^+ : \exists s, t \in \mathbb{Z}$ such that $\gcd(a, b) = sa + tb$
sa + tb, where s and t are integers.

$\gcd(a, b)$ can be expressed as a ***linear combination*** with integer coefficients of *a* and *b*.

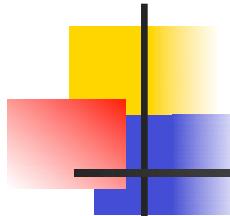
For example, $\gcd(6, 14) = 2$, and $2 = (-2) \cdot 6 + 1 \cdot 14$.

- Example:
 - Express $\gcd(252, 198) = 18$ as a linear combination of 252 and 198.



Theorem 1: Example

- Express $\gcd(252, 198) = 18$ as a linear combination of 252 and 198.
 - $252 = 1 \cdot 198 + 54$
 - $198 = 3 \cdot 54 + 36$
 - $54 = 1 \cdot 36 + 18$
 - $36 = 2 \cdot 18$
- $18 = 54 - 1 \cdot 36 = 54 - 1 \cdot (198 - 3 \cdot 54)$
$$\begin{aligned} &= 4 \cdot 54 - 1 \cdot 198 \\ &= 4 \cdot (252 - 1 \cdot 198) - 1 \cdot 198 \\ &= 4 \cdot 252 - 5 \cdot 198 \end{aligned}$$
- Therefore, $\gcd(252, 198) = 18 = 4 \cdot 252 + (-5) \cdot 198$



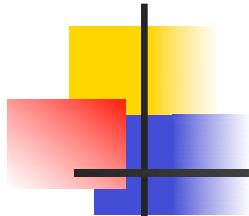
Proof of Lemma 1

■ Lemma 1:

$$\forall a, b, c \in \mathbb{Z}^+: \gcd(a, b) = 1 \wedge a|bc \rightarrow a|c$$

Proof:

- Applying theorem 1, $\exists s, t: sa + tb = 1$.
- Multiplying through by c , we have that $sac + tbc = c$.
- Since $a|bc$ is given, we know that $a|tbc$, and obviously $a|sac$.
- Thus (using the theorem on pp.202), it follows that $a|(sac + tbc)$; in other words, that $a|c$. ■

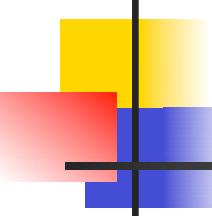


Proof of Lemma 2

- **Lemma 2:** If p is prime and $p|a_1a_2\dots a_n$ (integers a_i) then $p|a_i$ for some i .

Proof (by induction):

- If $n=1$, this is immediate since $p|a_0 \rightarrow p|a_0$.
Suppose the lemma is true for all $n < k$ and $p|a_1\dots a_k$.
- If $p|m$ where $m=a_1\dots a_{k-1}$ then we have it inductively.
- Otherwise, we have $p|ma_k$ but $\neg(p|m)$.
Since m is not a multiple of p , and p has no factors, m has no common factors with p , thus $\gcd(m,p)=1$.
So by applying Lemma 1, $p|a_k$. ■

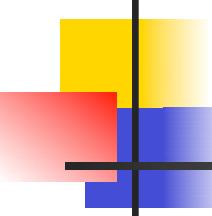


Theorem 2

- **Theorem 2:** Let $m \in \mathbb{Z}^+$ and $a, b, c \in \mathbb{Z}$.
If $ac \equiv bc \pmod{m}$ and $\gcd(c, m) = 1$, then $a \equiv b \pmod{m}$.

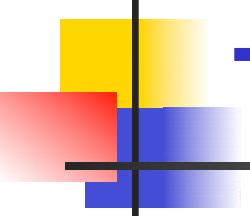
Proof:

- Since $ac \equiv bc \pmod{m}$, this means $m \mid ac - bc$.
 - Factoring the right side, we get $m \mid c(a - b)$.
Since $\gcd(c, m) = 1$, lemma 1 implies that $m \mid a - b$,
in other words, that $a \equiv b \pmod{m}$. ■
- Examples
 - $20 \equiv 8 \pmod{3}$ i.e. $5 \cdot 4 \equiv 2 \cdot 4 \pmod{3}$
Since $\gcd(4, 3) = 1$, $5 \equiv 2 \pmod{3}$
 - $14 \equiv 8 \pmod{6}$ but $7 \not\equiv 4 \pmod{6}$ (as $\gcd(2, 6) \neq 1$)



Linear Congruences, Inverses

- A congruence of the form $ax \equiv b \pmod{m}$ is called a ***linear congruence***. ($m \in \mathbb{Z}^+$, $a, b \in \mathbb{Z}$, and x : variable)
 - To *solve* the congruence is to find the x 's that satisfy it.
- An ***inverse of a , modulo m*** is any integer a^{-1} such that $a^{-1}a \equiv 1 \pmod{m}$.
 - If we can find such an a^{-1} , notice that we can then solve $ax \equiv b \pmod{m}$ by multiplying through by it, giving $a^{-1}ax \equiv a^{-1}b \pmod{m}$, thus $1 \cdot x \equiv a^{-1}b \pmod{m}$, thus $x \equiv a^{-1}b \pmod{m}$.

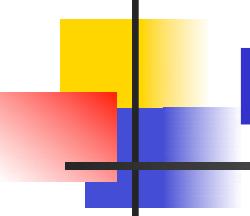


Theorem 3

- **Theorem 3:** If $\gcd(a,m)=1$ (i.e. a and m are relatively prime) and $m > 1$,
then a has a inverse a^{-1} unique modulo m .

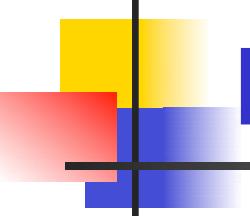
Proof:

- By theorem 1, $\exists s, t: sa + tm = 1$, so $sa + tm \equiv 1 \pmod{m}$.
 - Since $tm \equiv 0 \pmod{m}$, $sa \equiv 1 \pmod{m}$.
Thus s is an inverse of a (\pmod{m}).
 - Theorem 2 guarantees that if $ra \equiv sa \equiv 1$ then $r \equiv s$,
thus this inverse is unique modulo m .
(All inverses of a are in the same congruence class as s .)
-



Example

- Find an inverse of 3 modulo 7
 - Since $\gcd(3, 7) = 1$, by Theorem 3 there exists an inverse of 3 modulo 7.
 - $7 = 2 \cdot 3 + 1$
 - From the above equation, $-2 \cdot 3 + 1 \cdot 7 = 1$
 - Therefore, -2 is an inverse of 3 modulo 7
- Note that every integer congruent to -2 modulo 7 is also an inverse of 3, such as 5, -9 , 12, and so on.)



Example

- What are the solutions of the linear congruence $3x \equiv 4 \pmod{7}$?
 - -2 is an inverse of 3 modulo 7 (previous slide)
 - Multiply both side by -2 : $-2 \cdot 3x \equiv -2 \cdot 4 \pmod{7}$
 - $-6 \cdot x \equiv x \equiv -8 \equiv 6 \pmod{7}$
 - Therefore, the solutions to the congruence are the integers x such that $x \equiv 6 \pmod{7}$, i.e. $6, 13, 20, 27, \dots$ and $-1, -8, -15, \dots$

The Chinese Remainder Theorem

Let m_1, m_2, \dots, m_n be pairwise relatively prime positive integers greater than one and a_1, a_2, \dots, a_n arbitrary integers. Then the system

$$x \equiv a_1 \pmod{m_1},$$

$$x \equiv a_2 \pmod{m_2},$$

.

.

.

$$x \equiv a_n \pmod{m_n}$$

has a unique solution modulo $m = m_1 m_2 \cdots m_n$.

(That is, there is a solution x with

$0 \leq x < m$, and all other solutions are congruent modulo m to this solution.)



***Proof: To establish this theorem,
we need to show that a solution exists and that it is unique***
modulo m . We will show that a solution exists by describing a way
to construct this solution;

showing that the solution is unique modulo m

To construct a simultaneous solution,

first let $M_k = m/m_k$ for $k = 1, 2, \dots, n$. i.e., M_k is the product of the
moduli except for m_k .

Because m_i and m_k have no common factors greater than 1 when $i = k$, it follows that $\gcd(m_k, M_k) = 1$.

we know that there is an integer y_k , an inverse of M_k modulo m_k ,
such that $M_k y_k \equiv 1 \pmod{m_k}$.

To construct a simultaneous solution, form the sum
 $x = a_1 M_1 y_1 + a_2 M_2 y_2 + \dots + a_n M_n y_n$.



We will now show that x is a simultaneous solution.

First, note that because $Mj \equiv 0 \pmod{mk}$ whenever $j = k$, all terms except the k th term in this sum are congruent to 0 mod mk .

Because $Mkyk \equiv 1 \pmod{mk}$

$x \equiv akMkyk \equiv ak \pmod{mk}$, for $k = 1, 2, \dots, n$.

So x is a simultaneous solution to the n congruences.

What are the solutions of the systems of congruences

$$x \equiv 2 \pmod{3},$$

$$x \equiv 3 \pmod{5},$$

$$x \equiv 2 \pmod{7}?$$

let $m = 3 \cdot 5 \cdot 7 = 105$,

$M_1 = m/3 = 35$, $M_2 = m/5 = 21$, and $M_3 = m/7 = 15$.

We see that 2 is an inverse of $M_1 = 35$ modulo 3,

because $35 \cdot 2 \equiv 2 \cdot 2 \equiv 1 \pmod{3}$;

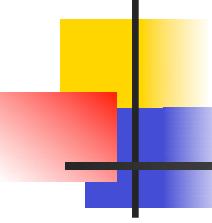
1 is an inverse of $M_2 = 21$ modulo 5, because $21 \equiv 1 \pmod{5}$;

1 is an inverse of $M_3 = 15$ ($\pmod{7}$), because $15 \equiv 1 \pmod{7}$.

The solutions to this system are those x such that

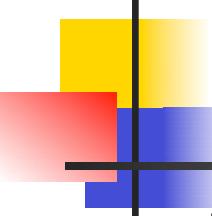
$$x \equiv a_1 M_1 y_1 + a_2 M_2 y_2 + a_3 M_3 y_3 = 2 \cdot 35 \cdot 2 + 3 \cdot 21 \cdot 1 + 2 \cdot 15 \cdot 1 = 233 \equiv 23 \pmod{105}.$$

23 is the smallest positive integer that is a simultaneous solution that leaves a remainder of 2 when divided by 3, a remainder of 3 when divided by 5, and a remainder of 2 when divided by 7.



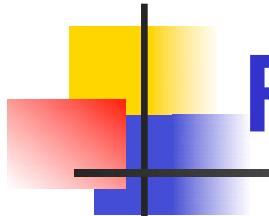
An Application of Primes!

- When you visit a secure web site ([https:...](https://) address, indicated by padlock icon in IE, key icon in Netscape), the browser and web site may be using a technology called *RSA encryption*.
- This *public-key cryptography* scheme involves exchanging *public keys* containing the product pq of two random large primes p and q (a *private key*) which must be kept secret by a given party.
- So, the security of your day-to-day web transactions depends critically on the fact that all known factoring algorithms are intractable!



Public Key Cryptography

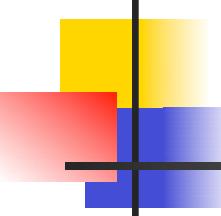
- In ***private key cryptosystems***, the same secret “key” string is used to both encode and decode messages.
 - This raises the problem of how to securely communicate the key strings.
- In ***public key cryptosystems***, there are two *complementary* keys instead.
 - One key decrypts the messages that the other one encrypts.
- This means that one key (the *public key*) can be made public, while the other (the *private key*) can be kept secret from everyone.
 - Messages to the owner can be encrypted by anyone using the public key, but can *only* be decrypted by the owner using the private key.
 - Like having a private lock-box with a slot for messages.
 - Or, the owner can encrypt a message with their private key, and then anyone can decrypt it, and know that *only* the owner could have encrypted it.
 - This is the basis of digital signature systems.
- The most famous public-key cryptosystem is RSA.
 - It is based entirely on number theory!



Rivest-Shamir-Adleman(RSA)

- Choose a pair p, q of large random prime numbers with about the same number of bits
 - Let $n = pq$
- Choose exponent e that is relatively prime to $(p-1)(q-1)$ and $1 < e < (p-1)(q-1)$
- Compute d , the inverse of e modulo $(p-1)(q-1)$.

- The **public key** consists of: n , and e .
- The **private key** consists of: n , and d .

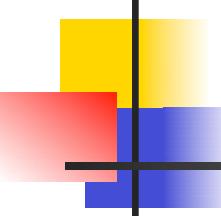


RSA Encryption

- To encrypt a message encoded as an integer:
 - Translate each letter into an integer and group them to form larger integers, each representing a block of letters. Each block is encrypted using the mapping

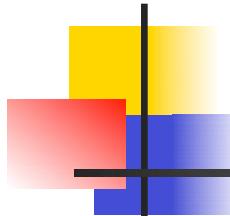
$$C = M^e \bmod n.$$

- Example: RSA encryption of the message **STOP** with $p = 43$, $q = 59$, and $e = 13$
 - $n = 43 \times 59 = 2537$
 - $\gcd(e, (p-1)(q-1)) = \gcd(13, 42 \cdot 58) = 1$
 - **STOP** -> 1819 1415
 - $1819^{13} \bmod 2537 = 2081$; $1415^{13} \bmod 2537 = 2182$
 - Encrypted message: **2081 2182**



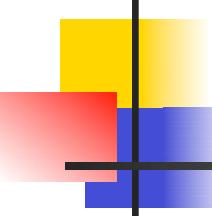
RSA Decryption

- To decrypt the encoded message C ,
 - Compute $M = C^d \bmod n$
 - Recall that d is an inverse of e modulo $(p-1)(q-1)$.
- Example: RSA decryption of the message **0981 0461** encrypted with $p = 43$, $q = 59$, and $e = 13$
 - $n = 43 \times 59 = 2537$; $d = 937$
 - $0981^{937} \bmod 2537 = 0704$
 - $0461^{937} \bmod 2537 = 1115$
 - Decrypted message: **0704 1115**
 - Translation back to English letters: **HELP**



Computer Arithmetic with Large Integers

- By Chinese Remainder Theorem, an integer a where $0 \leq a < m = \prod m_i$, $\gcd(m_i, m_{j \neq i}) = 1$, can be represented by a 's residues mod m_i : ($a \bmod m_1$, $a \bmod m_2$, ..., $a \bmod m_n$)
each m_i is an integer greater than 2, $\gcd(m_i, m_j) = 1$ whenever $i \neq j$,
 $m = m_1m_2\dots m_n$ is greater than the results of the arithmetic operations.
- To perform arithmetic with large integers represented in this way,
 - Simply perform operations on the separate residues!
 - Each of these might be done in a single machine operation.
 - The operations may be easily parallelized on a vector machine.
 - Works so long as $m >$ the desired result.



Computer Arithmetic Example

- For example, the following numbers are relatively prime:

$$m_1 = 2^{25}-1 = 33,554,431 = 31 \cdot 601 \cdot 1,801$$

$$m_2 = 2^{27}-1 = 134,217,727 = 7 \cdot 73 \cdot 262,657$$

$$m_3 = 2^{28}-1 = 268,435,455 = 3 \cdot 5 \cdot 29 \cdot 43 \cdot 113 \cdot 127$$

$$m_4 = 2^{29}-1 = 536,870,911 = 233 \cdot 1,103 \cdot 2,089$$

$$m_5 = 2^{31}-1 = 2,147,483,647 \text{ (prime)}$$

- Thus, we can uniquely represent all numbers up to

$m = \prod m_i \approx 1.4 \times 10^{42} \approx 2^{139.5}$ by their residues r_i modulo these five m_i .

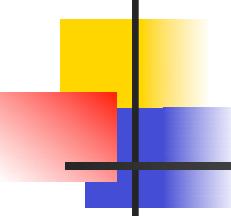
- E.g., $10^{30} = (r_1 = 20,900,945; r_2 = 18,304,504; r_3 = 65,829,085;$
 $r_4 = 516,865,185; r_5 = 1,234,980,730)$

- To add two such numbers in this representation,

- Just add the residues using machine-native 32-bit integers.
 - Take the result mod 2^k-1 :

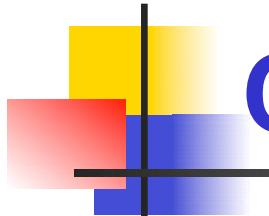
- If result is \geq the appropriate 2^k-1 value, subtract out 2^k-1
 - or just take the low k bits and add 1.

- Note: No carries are needed between the different pieces!



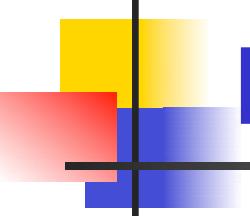
Pseudoprimes

- Ancient Chinese mathematicians noticed that whenever n is prime, $2^{n-1} \equiv 1 \pmod{n}$.
 - Some also claimed that the converse was true.
- However, it turns out that the converse is not true!
 - If $2^{n-1} \equiv 1 \pmod{n}$, it doesn't follow that n is prime.
 - For example, $341 = 11 \cdot 31$, but $2^{340} \equiv 1 \pmod{341}$.
- Composites n with this property are called *pseudoprimes*.
 - More generally, if $b^{n-1} \equiv 1 \pmod{n}$ and n is composite, then n is called a *pseudoprime to the base b*.



Carmichael Numbers

- These are sort of the “ultimate pseudoprimes.”
- A *Carmichael number* is a composite n such that $b^{n-1} \equiv 1 \pmod{n}$ for all b relatively prime to n .
- The smallest few are 561, 1105, 1729, 2465, 2821, 6601, 8911, 10585, 15841, 29341.



Fermat's Little Theorem

- Fermat generalized the ancient observation that $2^{p-1} \equiv 1 \pmod{p}$ for primes p to the following more general theorem:
- **Theorem:** (Fermat's Little Theorem.)
 - If p is prime and a is an integer not divisible by p , then $a^{p-1} \equiv 1 \pmod{p}$.
 - Furthermore, for every integer a we have $a^p \equiv a \pmod{p}$.
- Example (Exponentiation MOD a Prime)
 - Find $2^{301} \pmod{5}$: By FLT, $2^4 \equiv 1 \pmod{5}$. Hence,
 $2^{300} = (2^4)^{75} \equiv 1 \pmod{5}$.
Therefore, $2^{301} = (2^{300}) \cdot 2 \equiv 1 \cdot 2 \equiv 2 \pmod{5}$