<u>Chapter 1</u>

1 Propositional Logic:

- **Propositional Logic (Calculus):** It deals with propositions.
- **Proposition:** A statement that is either **<u>true</u>** or **<u>false</u>** (but not both).
- **Examples of propositions:**
  - Today is Sunday. (True) since, today is Sunday.
  - Today it is raining. (True) since, today is raining.
  - 1 + 1 = 4. (False), since 1 + 1 = 2
  - Water boils at 50°C. (False) → Scientific Fact
  - Ali has a cat. (True) as example for a story.
  - Ali has a dog. (false) as example for a story.

- **Examples of non-propositions:**
  - 1 + x = 4. Since x is a variable
  - Close the door. (order)
  - What is your name? (question)
  - Wow!!!!! (Exclamation)

Propositions can be denoted by **Letters.**
- **True** value can be denoted by **T.**
- **False** value can be denoted by **F.**
- **Example:**
  - **P:** Today is Friday. : **T**
  - **Q:** 1 + 1 = 4. : **F**

- **Propositions** can be:
  1. **Atomic:** consists of single proposition.
  2. **Compound:** consists of one or more propositions connected by logical operators.

- **Example:**
  - **P:** Today is Friday. : **T** **Atomic**
  - **Q:** 1 + 1 = 4. : **F** **Atomic**
  - **R:** $P \wedge Q$: **F** **Compound**

<u>Truth Table</u>

- A ***Truth Table*** is a complete list of the possible truth values of a logical statement.
- Truth table can be used to show the effect of each logical operator, and it can be also used to show the result of a logical statement.

- **Logical Operators:**

Assume that P, Q, and R are propositions

1- Negation: for **P**, Negation of **P** is denoted by **~P,** and it is read as **"NOT P"**
Negation reverses the truth value of P.

- **P:** Today is Friday. : **T** **Atomic**
- **Q:** $1 + 1 = 4$. : **F** **Atomic**
- **~P:** Today is not Friday. : **F** **compound**
- **~Q:** $1 + 1 \neq 4$. : **T** **compound**

- Truth Tables of a single proposition **P** or its Negation **~P**:

P
T
F

P	**~P**
T	F
F	T

2- Conjunction: is denoted by **P ∧ Q,** and it is read as **"P AND Q"**
Conjunction is True, if <u>both</u> P and Q are true.

Let:
P: Ali has a cat.
Q: Ali has a dog
P ∧ Q : Ali has a cat <u>and</u> a dog. It is True when Ali has 2 pets both are a cat and a dog

P	Q	P ∧Q
T	**T**	**T**
T	**F**	**F**
F	**T**	**F**
F	**F**	**F**

3- Disjunction:

a. inclusive OR is denoted by **$P \vee Q$,** and it is read as **"P OR Q"**
It is True, if <u>any</u> of P and Q is true.

Let:
P: Ali has a cat.
Q: Ali has a dog
$P \vee Q$: Ali has a cat <u>or</u> a dog. It is <u>True</u> when Ali has a cat or a dog or both.

P	Q	$P \vee Q$
T	T	T
T	F	T
F	T	T
F	F	F

b. exclusive OR is denoted by **$P \oplus Q$,** and it is read as **"P XOR Q"**
It is True, if <u>any</u> of P and Q is true, but not both. i.e. if they are different.

Let:
P: Ali has a cat.
Q: Ali has a dog
$P \oplus Q$: Ali has one pet, Ali has a cat <u>or</u> a dog.
It is <u>True</u> when Ali has a cat or a dog <u>but not both</u>.

Let:
R : Ahmad is tall.
S : Ahmad is short.
W: Ahmad is fat.
$R \oplus S$: Ahmad is tall or short.
$R \vee W$: Ahmad is tall or fat.

P	Q	$P \oplus Q$
T	T	F
T	F	T
F	T	T
F	F	F

4- Implication: is denoted by **$P \rightarrow Q$,** and it is read as **"P implies Q"**

It is false, only if P is T and Q is F

$P \rightarrow Q$ has many forms in English Language:

" If P, then Q"	"If P, Q"	"P only if Q"
"P implies Q"	"Q if P"	"Q unless ~P"
"When P, then Q"	"Whenever P, Q"	

P	Q	$P \rightarrow Q$
T	T	T
T	F	F
F	T	T
F	F	T

USING:
- **P:** it rains.
- **Q:** I wear my coat.
- **P → Q** : has many forms:
  1. **If it rains, then I will wear my coat.**
  2. **If it rains, I will wear my coat.**
  3. **It rains only if I wear my coat.**
  4. **Raining implies that I will wear my coat.**
  5. **I will wear my coat, if it rains.**
  6. **I will wear my coat unless it is not raining.**
  7. **Unless it is not raining, I will wear my coat.**

5- Biconditional: is denoted by **P↔Q,** and it is read as **"P if and only if Q"**

It is true, if P and Q both have the same truth value.

P ↔ Q has many forms in English Language:

"P if and only if Q"

"If P, then Q, and conversely"

"P is sufficient and necessary for Q"

USING:

- **P:** it rains.
- **Q:** I wear my coat.
- **P** ↔ Q : has many forms:
    1. **If and only if it rains, I will wear my coat.**
    2. **If it rains, I will wear my coat, and conversely.**
    3. **If it rains, I will wear my coat and if I wear my coat, it will rain**

P	**Q**	**P ↔ Q**
T	T	T
T	F	F
F	T	F
F	F	T

P ↔ Q is the same as:

$$(p \rightarrow Q) \wedge (Q \rightarrow P)$$

Examples:

USING:

- **P:** Samer has a car.
- **Q:** Samer has a bicycle.
- **R:** Today is sunny.
- **S:** It rains.
- **W:** I wear my umbrella.

WE CAN BUILD:

- **~R:** Today is not sunny.
- **$P \wedge Q$**: Samer has a car and a bicycle.
- **$P \vee Q$**: Samer has a car or a bicycle.
- **$P \oplus Q$**: Samer has a divining machine; it is either a car or a bicycle.
- **$S \rightarrow W$:** If it rains, I will wear my umbrella.
- **$S \leftrightarrow W$:** If it rains, I will wear my umbrella, and conversely.

- **The following truth table is used to represent the compound proposition:**

$(P \wedge Q) \vee (\sim P)$

P	**Q**	**$P \wedge Q$**	**~P**	**$(P \wedge Q) \vee (\sim P)$**
T	T	T	F	T
T	F	F	F	F
F	T	F	T	T
F	F	F	T	T

Note: If a compound proposition has ***n*** distinct simple components, then it will have **2^n** rows in its truth table, as this is the number of possible combinations of n components, each with 2 possible truth values **T** or **F**.

- ***P*** **→ *Q*** **has 3 components: Converse, contrapositive, Inverse**

Assume: **(*P* → *Q*)** **if it is raining, then it is cloudy.**

P **Q**

1. Converse	**Q→P**	**If it is cloudy, then it is raining**
2. Contrapositive	**¬Q → ¬P**	**If it is not cloudy, then it is not raining**
3. Inverse	**¬p→¬Q**	**if it is not raining, then it is not cloudy**

- **Logical operator Precedence**

Operator	**Precedence**
¬	**1**
∧ ∨ ⊕	**2** **3** **4**
→ ↔	**5** **6**

Ex: **Assume P**: T **Q**: F **R**: F

Find the value of:

$P \vee Q \wedge \neg R \leftrightarrow P$

$T \vee F \wedge \neg F \leftrightarrow T$

$T \vee F \wedge T \leftrightarrow T$

$T \vee F \leftrightarrow T$

$T \leftrightarrow T$

T

- **Translation into English Sentences**

1. if <u>you are a computer science major</u> or <u>you are not a freshman</u>, <u>you can access the internet in the lab</u>.

P ¬ Q R

$(P \vee \neg Q) \rightarrow R$

2. If <u>you watch television</u> <u>your mind will decay</u>, and conversely.

P Q

$P \leftrightarrow Q$

3. <u>You got an A in this class</u>, but <u>you did</u> not <u>do every exercise in the book</u>.

P Q

$P \wedge \neg Q$

4. if <u>it is hot outside</u> <u>buy an ice cream</u>, and if <u>you buy an ice cream</u> <u>it is hot outside</u>.

P Q Q P

$(P \rightarrow Q) \wedge (Q \rightarrow P) \equiv P \leftrightarrow Q$

5. <u>You got an A in this class</u>, only if <u>you do every exercise in the book</u>.

P Q

$P \rightarrow Q$

6. <u>You got an A in this class</u>, if <u>you do every exercise in the book</u>.

P Q

$Q \rightarrow P$

7. <u>You will</u> not <u>got an A in this class</u>, unless <u>you did do every exercise in the book</u>.

P Q

$\neg Q \rightarrow \neg P$

- **Logical And Bit Operations**

- **Bit** has two values: 0, 1
- True (1), False (0)
- **Boolean Variable:** a variable that is either true or false.
- **Bit operation** corresponds to logical connectives:

Logical Operator	**Bit operator**
$\neg$	**NOT**
$\wedge$ $\vee$ $\oplus$	**AND** **OR** **XOR**

- **Bit string:** it is a sequence of zero or more bits.
- **String Length:** number of bits in the Bit string.

Ex1: 101010011 is a bit string with **length = 9**
Ex2:

01 1011 0110 11 0001 1101 **OR**	01 1011 0110 11 0001 1101 **AND**	01 1011 0110 11 0001 1101 **XOR**
11 1011 1111	01 0001 0100	10 1010 1011

NOT (01 1011 0110) = 10 0100 1001

2 Logical equivalence

Def: 1. Tautology: compound proposition that is always true (Ex: $P \vee \neg P$)
2. Contradiction: compound proposition that is always false (Ex: $P \wedge \neg P$)
3. Contingency: compound proposition that is either true or false (Ex: $P \rightarrow Q$)

- **Logical Equivalence ($P \equiv Q$, $P \Leftrightarrow Q$)**

Def: the two compound propositions **P,Q** are logically equivalent if $P \leftrightarrow Q$ is a tautology .

A. Using truth table

Ex1: show that $P \rightarrow Q \equiv \neg P \vee Q$

P	Q	$\neg P$	$P \rightarrow Q$	$\neg P \vee Q$	$P \rightarrow Q \leftrightarrow \neg P \vee Q$
T	**T**	**F**	**T**	**T**	**T**
T	**F**	**F**	**F**	**F**	**T**
F	**T**	**T**	**T**	**T**	**T**
F	**F**	**T**	**T**	**T**	**T**

It is a Tautology
$\therefore$ **They are equivalent**

Ex2: show that $\neg(P \vee Q) \equiv (\neg P \wedge \neg Q)$ using truth table

B. Using Logical equivalence rules

Table 1: Equivalence rules: $\wedge \vee \neg$

Eq	
P ∧ P ∨	
P P	on
P P	nt
¬ ¬	
$\neg(\neg P) \equiv P$	Double Negation
$P \vee Q \equiv Q \vee P$ P ∧	Commutative
(P (P	sociative
P ∨ R) P ∧ ∧R	tributive
$\neg (P \vee Q) \equiv \neg P \wedge \neg Q$ $\neg (P \wedge Q) \equiv \neg P \vee \neg Q$	Demorgan's
$P \vee (P \wedge Q) \equiv P$ $P \wedge (P \vee Q) \equiv P$	Absorption

Table 2: Implications Logical Rules

1. $P \rightarrow Q \equiv \neg P \vee Q$
2. $P \rightarrow Q \equiv \neg Q \rightarrow \neg P$
3. $P \vee Q \equiv \neg P \rightarrow Q$
4. $P \wedge Q \equiv \neg(P \rightarrow \neg Q)$
5. $\neg(P \rightarrow Q) \equiv P \wedge \neg Q$
6. $(P \rightarrow Q) \wedge (P \rightarrow R) \equiv P \rightarrow (Q \wedge R)$
7. $(Q \rightarrow R) \wedge (P \rightarrow R) \equiv (Q \vee P) \rightarrow R$
8. $(P \rightarrow Q) \vee (P \rightarrow R) \equiv P \rightarrow (Q \vee R)$
9. $(Q \rightarrow R) \vee (P \rightarrow R) \equiv (Q \wedge P) \rightarrow R$

Table 3: Bicondintional Rules

$P \leftrightarrow Q \equiv (P \rightarrow Q) \wedge (Q \rightarrow P)$

$P \leftrightarrow Q \equiv \neg P \leftrightarrow \neg Q$

$P \leftrightarrow Q \equiv (P \wedge Q) \vee (\neg P \wedge \neg Q)$

$\neg(P \leftrightarrow Q) \equiv P \leftrightarrow \neg Q$

Ex1: show that $(P \wedge Q) \rightarrow (P \vee Q)$ is a tautology

1.	$\neg (P \wedge Q) \vee (P \vee Q)$	Implication rule
2.	$(\neg P \vee \neg Q) \vee (P \vee Q)$	Demorgan's Law
3.	$(\neg P \vee P) \vee (\neg Q \vee Q)$	Associative and commutative
4.	$T \vee T$	Negation law
5.	T	

Ex2: show that $\neg (P \vee (\neg P \wedge Q))$ and $(\neg P \wedge \neg Q)$ are logically equivalent.

1.	$\neg (P \vee (\neg P \wedge Q) \equiv \neg P \wedge \neg (\neg P \wedge Q)$	Demorgan's
2.	$\equiv \neg P \wedge (\neg(\neg P) \vee \neg Q)$	Demorgan's
3.	$\equiv \neg P \wedge (P \vee \neg Q)$	Double negation
4.	$\equiv (\neg P \wedge P) \vee (\neg P \wedge \neg Q)$	Distributive
5.	$\equiv F \vee (\neg P \wedge \neg Q)$	Negation
6.	$\equiv (\neg P \wedge \neg Q)$	Identity

3. PREDICATES AND QUANTIFIERS

PREDICATES

- **$X > 3$**
- **$X = Y + 3$**

  Both above statements are not propositions, they are called <u>predicates</u>

<u>Ex1:</u>
$P(x)$: $x > 3$
$P(2)$: $2 > 3$: F
$p(4)$: $4 > 3$: T
It is called: Propositional function

<u>Ex2:</u> $Q(x, y)$: $x = y+3$ $\quad$ $Q(3,0)$: $3 = 0 + 3$: T

<u>Ex3:</u> $X + Y = Z$. $\quad$ $R(X,Y,Z)$: $X+Y = Z$ $R(2,3,4)$: $2+3 = 4$:F

<u>QUANTIFIERS</u>

Quantifiers → Universal quantifier ($\forall$), for all
↘ Existential Quantifier ($\exists$), for some

1. <u>Universal Quantifier</u>

*** $P(x)$ is true for all values of x in the <u>universe of discourse</u> (domain). ➔ $\forall x\ p(x)$**
* **$\forall x\ p(x)$ is read as : "** for all x p(x) " , " for every x p(x)"

$\forall x\ p(x)$ $\equiv$ $p(e1) \wedge p(e2) \wedge p(e3) \wedge \ldots \wedge p(en)$, where $\{e1,e2,\ldots,en\}$ are all elements of the domain
$\forall x\ p(x)$: T , if p(x) is true for **<u>all elements</u>**

<u>Ex1:</u> $p(x)$: "$x+1 > x$" , what is the truth value of $\forall x\ p(x)$, where the domain is all real numbers?

Sol : $\forall x\ p(x)$ is true for all values of x

<u>Ex2:</u> What is the truth value $\forall x\ p(x)$, where $p(x)$ is ($x*x < 10$). The domain is all positive integers not exceeding 4?

Sol : $P(1) \wedge p(2) \wedge p(3) \wedge p(4)$
$T \wedge T \wedge T \wedge F = F$

<u>Ex3:</u> what is the truth value of $\forall x\ (x*x \geq x)$, if the domain is all integer numbers

Sol: T

<u>Ex4:</u> Translate the following statement into English language:
$\forall x\ Q(x)$, where Q(x) is "x has two parents" and the domain is all people.

Sol: every person has two parents

2. <u>Existential quantifier</u>

- **There exists an element x in the domain such that p(x) is true ➔ $\exists x\ p(x)$**
- **$\exists x\ p(x)$ is read as: "there is a x such that p(x) "," there is at least one x such that p(x)"**

$\exists x\ p(x)$ $\equiv$ $p(e1) \vee p(e2) \vee p(e3) \vee \ldots.. \vee p(en)$, where {e1,e2,…,en} are all elements of the domain
$\exists x\ p(x)$: T , if p(x) is true for **<u>at least one element</u>**

<u>Ex1:</u> what is the truth value of $\exists x\ p(x)$, where p(x) is " x*x > 10" and the domain is all integers not exceeding 4?

$\exists x\ p(x) = P(1) \vee p(2) \vee p(3) \vee p(4)$ = True, since p(4) is True

<u>Ex2:</u> p(x): x> 1 what is the truth value of $\exists x\ p(x)$, where the domain is all real numbers?

Ans: True

Binding Variable

A variable in a predicate might be:

1- **Free:**

Ex1: p(x): x has a cat. Domain: people x is a free variable.

Ex2: like(x, y): x likes y. Domain: people x and y are free variables.

2- **Bound:**

a. **To a value**

Ex1: p(Ali): Ali has a cat. x is a bound variable to value Ali.

Ex2: like(Ali, Ahmad): Ali likes Ahmad. x and y are bound variables to values.
like(Ali, y): Ali likes y. x is a bound variable to a value, y is free.==> it is a predicate.

b. **To a quantifier**

Ex: $\forall x \exists y$ like(x,y) x is bound to $\forall$, y is bound to $\exists$
Ex: $\exists x$ Q(x, y) x is bound, y is free
Ex: $\exists x\ (p(x) \wedge Q(x)) \quad \vee \quad \forall x\ R(x)$
- x is bound to $\exists$**x**, - x is bound to $\forall$**x**
- Scope of $\exists$**x** is **$(p(x) \wedge Q(x))$** - scope of $\forall$**x** is **R(x)**

This statement can be written as :
$\exists x\ (p(x) \wedge Q(x)) \vee \forall y\ R(y)$
But if it becomes : $\exists x\ (p(x) \wedge Q(x)) \vee R(y)$
since y is free, so this is a predicate (not a proposition)
because a proposition might be a predicate with no free variables.
so ( ) following the quantifier specified the scope of it. if there is no ( ), the scope of the quantifier will be the first predicate only. Like:
$\exists x\ p(x) \wedge Q(x) \vee \forall y\ R(y) \equiv \exists x\ p(x) \wedge Q(z) \vee \forall y\ R(y)$
Because Q is out of $\exists$ scope.

Negation

1. $\neg \forall x\ P(x) \equiv \exists x \neg p(x)$
ex: Every student in the class has taken calculus. $\forall x\ P(x)$
There is a student in the class who has not taken Calculus. $\neg \forall x\ P(x) \equiv \exists x \neg p(x)$

2. $\neg \exists x\ P(x) \equiv \forall x \neg p(x)$
ex: There is a student in the class who has taken Calculus $\exists x\ p(x)$
Every student in the class has not taken calculus. $\neg \exists x\ P(x) \equiv \forall x \neg p(x)$

Ex: what are the negations of the following statements?
A. $\forall x\ (x*x > x)$
Sol: $\neg \forall x\ (x*x > x) \rightarrow \exists x \neg (x*x > x) \rightarrow \exists x\ (x*x \leq x)$

B. $\exists x\ (x*x = 2)$
Sol : $\neg \exists x\ (x*x = 2) \rightarrow \forall x \neg (x*x = 2) \rightarrow \forall x\ (x*x \neq 2)$

4 NESTED QUANTIFIERS

Ex1: $\forall x \forall y\ (x + y = y + x)$ is true, for every values x and y $x + y = y + x$ <u>Domain: Real Numbers.</u>

Ex2: $\forall x \forall y\ (x + y = 0)$ is false for every values x and y $x + y = 0$ <u>Domain: Real Numbers.</u>

Ex3: C(x) is "x has a computer"
F(x, y) is "x and y are friends

Translate the statement:
$\forall x\ (C(x) \vee \exists y\ (C(y) \wedge F(x, y)))$

Sol: For every student x in your school x has a computer or there is a student y such that y has a computer and x and y are friends.

Or

Every student in your school has a computer or has a friend ho has a computer

NEGATING NESTED QUANTIFIER

Ex: $\neg \forall x \exists y\ (xy = 1) \rightarrow \exists x \neg \forall y\ (xy = 1) \rightarrow \exists x \forall y\ (xy \neq 1)$
Ex: $\neg \forall x \forall y\ \exists z\ (P(x,y) \wedge Q(y,z)) \rightarrow \exists x \neg \forall y\ \exists z\ (P(x,y) \wedge Q(y,z))$
$\rightarrow \exists x\ \exists y\ \neg \exists z\ (P(x,y) \wedge Q(y,z)) \rightarrow \exists x\ \exists y\ \forall z \neg\ (P(x,y) \wedge Q(y,z))$
$\rightarrow \exists x\ \exists y\ \forall z\ (\neg P(x,y) \vee \neg Q(y,z))$

ORDER OF QUANTIFIER

Statement	**When true**	**When false**
$\forall x \forall y\ P(x, y)$ **$\forall y \forall x\ P(x, y)$**	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 x, such that P(x, y) is false
$\exists x \forall y\ P(x, y)$	There is 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.

Using Domain: All integers

Ex1: Let Q(x, y) denote " $x + y = 0$" what are the truth values of the quantifications $\exists x \forall y\, Q(x, y)$, and $\forall x \exists y\, Q(x, y)$?

Sol: $\exists x \forall y\, Q(x, y)$ false $\forall x \exists y\, Q(x, y)$ true

Ex2: Let Q(x, y) denote " $x + y = x$" what are the truth values of the quantifications $\exists y \forall x\, Q(x, y)$, and $\forall x \exists y\, Q(x, y)$?

Sol: $\exists y \forall x\, Q(x, y)$ true $\forall x \exists y\, Q(x, y)$ true

Ex3: Let Q(x, y) denote " $x + y = y + x$" what are the truth values of the quantifications $\exists y \forall x\, Q(x, y)$, and $\forall x \forall y\, Q(x, y)$?

Sol: $\exists y \forall x\, Q(x, y)$ true $\forall x \forall y\, Q(x, y)$ true

Ex4: Let Q(x, y) denote " $x + y = 5$" what are the truth values of the quantifications $\exists y \exists x\, Q(x, y)$, and $\forall x \forall y\, Q(x, y)$?

Sol: $\exists y \exists x\, Q(x, y)$ true $\forall x \forall y\, Q(x, y)$ false

Ex5: Let Q(x, y) denote " $x + y = 0.5$" what are the truth values of the quantifications $\exists y \exists x\, Q(x, y)$, and $\forall x \forall y\, Q(x, y)$?

Sol: $\exists y \exists x\, Q(x, y)$ false $\forall x \forall y\, Q(x, y)$ false

Translation From English Into Logical Expressions

Examples:

1. **Express the statement :**
"For every person x, if person x is a student in this class then x has studied Calculus".
Domain: All people **S(x)** **C(x)**
$\forall x\ (S(x) \rightarrow C(x))$

2. **Express the statement:**
"Every student x, if x is a student in this class then x has studied Calculus".
Domain: All students in this class. **C(x)**

$\forall x\ C(x)$

3. "No one is perfect"
$\forall x \neg P(x)$

4. "All your friends are perfect."
F(x): your friend **P(x): perfect**
$\forall x\ (F(x) \rightarrow P(x))$

5. Let P(x) be the statement "x can speak French" and Q(x) be the statement "x knows C++". The domain is all students in the school. Express the following statement using quantifiers and logical operator:

A. "No student at your school can speak French or knows C++."

$\forall x \neg (P(x) \vee Q(x))$

B. There is a student at your school who can speak French but does not know C++.
$\exists x\ (P(x) \wedge \neg Q(x))$

6. Express the statement "if a person is a female and is a parent, then this person is someone's mother"

F(x): person is a female
P(x): person is a parent
M(x, y): x is the mother of y

Sol: $\forall x\ ((F(x) \wedge P(x)) \rightarrow \exists y\ M(x, y))$

7. Express the statement: "Everyone has one best friend"
Sol:
B(x, y) : x is the best friend of y
$\forall x \exists y\ B(x, y)$

Chapter 1 Exercise on Translation

- **Exercise 1:**

Domain: people
Teacher(x): x is a teacher.
Student(x): x is a student.
Visit(x, y): x visited y.
Translate the following into Logic:

A. Ali visited Sami.
B. Ali visited everyone.
C. Ali visited someone.
D. Ali visited some teachers.
E. Ali visited all teachers.
F. Someone visited someone.
G. Everyone visited someone.
H. Someone visited everyone.
I. Everyone visited everyone.
J. Everyone has been visited by someone.
K. Ali did not visit anyone ⇔ Ali visited nobody.
L. Ali did not visit everyone.
M. All students visited Ali and some teacher too.
N. All students visited Ali and some teacher did.
O. Ali visited everyone but nobody visited him.

- **Exercise 2:**

Let:
Domain: Animals

Translate the following into Logic:
Domain: Animals

A. All animals have skin.
B. All dogs have legs.
C. Some cats are black.
D. Some cats are black or white.
E. No animal can speak English.
F. If there is an animal, then it has a mother.

- **Exercise 3:**

Let:
Domain1: people
Domain2: fruits.
L(x, y): x likes y.
Friend (x, y): x is a friend of y.
Student(x): x is a student.
Teacher(x): x is a teacher.
Teach(x, y): x teaches y.

Translate the following into Logic:

A. Everybody likes apples.
B. Somebody likes apples but not oranges.
C. Everybody likes apples or oranges.
D. Everybody likes some fruits.
E. Everybody likes somebody.
F. Everyone likes Ali.
G. Ahmad likes Ali.
H. Someone likes every one.
I. No one likes every one.
J. Everyone likes himself/herself.
K. There is someone whom everybody likes.
L. Some students like some teachers.
M. Ali and Ahmad are friends.
N. Some students are friends.
O. Every teacher has taught Ali.
P. Some teachers have taught Ali and all his friends.
Q. Ali has a friend who has been taught by all teachers.
R. Some teachers have taught all students.
S. Some students and some teachers are friends.
T. If someone is a teacher, then Ali likes him.
U. If a person is a teacher, then he taught some students.

5 Rules of Inference

Example:

P : T **(hypothesis, or premise)**

$P \rightarrow Q$: T **(hypothesis, or premise)**

$\therefore Q$: T **(Conclusion)**

(Therefore)

Rules of inference	Tautology	Name
P **------** $\therefore P \vee Q$	$P \rightarrow (p \vee Q)$	**Addition**
$P \wedge Q$ **------** $\therefore P$	$(P \wedge Q) \rightarrow P$	**Simplification**
P **Q** **------** $\therefore P \wedge Q$	$((p) \wedge (Q)) \rightarrow (P \wedge Q)$	**Conjunction**
P $P \rightarrow Q$ **------** $\therefore Q$	$[P \wedge (P \rightarrow Q)] \rightarrow Q$	**Modus ponens**
$\neg Q$ $P \rightarrow Q$ **------** $\therefore \neg P$	$[\neg Q \wedge (P \rightarrow Q)] \rightarrow \neg P$	**Modus Tollens**
$P \rightarrow Q$ $Q \rightarrow R$ **--------** $\therefore P \rightarrow R$	$[(p \rightarrow Q) \wedge (Q \rightarrow R)] \rightarrow (P \rightarrow R)$	**Hypothetical Syllogism**
$P \vee Q$ $\neg P$ **--------** $\therefore Q$	$[(P \vee Q) \wedge \neg P] \rightarrow Q$	**Disjunctive syllogism**
$P \vee Q$ $\neg P \vee R$ **----------** $\therefore Q \vee R$	$[(P \vee Q) \wedge (\neg P \vee R)] \rightarrow Q \vee R$	**Resolution**

<u>Ex:</u> state the rule of inference for**: "It is below freezing now, therefore it is either below freezing or raining now"**

Sol:

It is below freezing: P
It is raining: Q

P

$\therefore$ P v Q

It is called: Addition Inference Rule.

<u>Ex:</u> State the rule of inference used in the argument
"If it is rain today, then we will not barbecue today". "If we don't barbecue today then we will have a barbecue tomorrow". Therefore, "if it rains today, then we will have a barbecue tomorrow".

Sol:
If it is rain today: **P** we will barbecue today: **Q**
We will have barbecue tomorrow: **R**

1. $P \rightarrow \neg Q$
2. $\neg Q \rightarrow R$

$\therefore P \rightarrow R$ using H. S. of step 1 and step 2

<u>Valid Argument</u>

- An Argument form is called **valid** if whenever the entire hypothesizes are true, the conclusion is also true.
- Consequently Q logically follows from the hypothesis p1,p2, p3, ….., pn:

$(P1 \wedge P2 \wedge P3 \ldots\ldots \wedge Pn) \rightarrow Q$

EX: Show that the hypothesis “It is not sunny this afternoon and it is colder than yesterday.” “we will go swimming only if it is sunny,” “ if we do not go swimming, then we will take a canoe trip,” and “ If we take a canoe trip, then we will be home by sunset” leads to the conclusion “ we will be home by sunset”

P: It is sunny this afternoon
Q: it is colder than yesterday
R: we will go swimming only if it is sunny
S: we will take a canoe trip
H: we will be home by sunset

Hypothesis
1. $\neg P \wedge Q$
2. $R \rightarrow P$
3. $\neg R \rightarrow S$
4. $S \rightarrow H$

Solution:

Step	**Reason**
1. $\neg P \wedge Q$	**Hypothesis**
2. $\neg P$	**Simplification using step 1**
3. $R \rightarrow P$	**Hypothesis**
4. $\neg R$	**Modus tollens step 2 + 3**
5. $\neg R \rightarrow S$	**Hypothesis**
6. S	**modus ponens using step 4 and 5**
7. $S \rightarrow H$	**Hypothesis**

$\therefore$ **H**	**Conclusion (modus ponense using step 6 and 7**

Fallacies

There are two types of fallacies:
A. Fallacy of affirming the conclusion $[(P \rightarrow Q) \wedge Q] \rightarrow P$

This may be wrong you may get an A without solving every problem in the book.

B. Fallacy of Denying the Hypothesis $[(P \rightarrow Q) \wedge \neg P] \rightarrow \neg Q$

Rules Of Inference For Quantified Statements

Rules Of Inference	Name
$\forall x P(x)$ --------- $\therefore$ P(c) for **all** elements c	Universal Instantiation
P(c) for **every** element c --------- $\therefore \forall x P(x)$	Universal Generalization
$\exists x P(x)$ --------- $\therefore$ P(c) for some element c	Existential Instantiation
P(c) for some element c ---------- $\therefore \exists x P(x)$	Existential Generalization

EX1: show that the premises " Everyone in this class has taken a course in computer science" and " Marla is a student in this class" imply the conclusion " Marla has taken a course in computer science".

D(x) : x in this class
C(x) : x has taken a course in computer science.

Sol:
1. $\forall x(D(x) \rightarrow C(x))$ Premise #1
2. D(Marla) → C(Marla) Universal instantiation from 1
3. D(Marla) Premise #2

$\therefore$ C(Marla) Modus Ponens from 2 and 3

EX2: show that the premises "Everyone in this class has taken a course in computer science" and "Someone is a student in this class" imply the conclusion "Someone has taken a course in computer science".

D(x) : x in this class
C(x) : x has taken a course in computer science.

Sol:
1. $\forall x(D(x) \rightarrow C(x))$ Premise #1
2. D(a) → C(a) Universal instantiation from 1
3. $\exists$x D(x) Premise #2
4. D(a) Existential instantiation from 3
5. C(a) Modus Ponens from 2 and 4

$\therefore$ $\exists$x C(x) Existential generalization from 5

6. Introduction to Proofs

Methods Of proofs

1. Direct Proof

- The implication $p \rightarrow Q$ can be proved by showing that if P is true then Q must also be true.
- Integer **n** is even if there exists an integer K such that n=2K
- Integer **n** is Odd if there exists an integer K such that n=2K+1

Ex: Give a direct proof of the theorem "if n is an odd integer, then n^2 is an odd integer"

Sol:

1. Assume n is odd. n = 2K+1
2. It follows that $n^2 = (2K+1)^2 = 4K^2+4K + 1 = 2(2K^2+2K) + 1$ is also odd
5. therefore n^2 is odd

2. Indirect Proof

- The implication $p \rightarrow Q$ is equivalent to it's contrapositive $\neg Q \rightarrow \neg P$
- To prove that $p \rightarrow Q$ is true we should prove that $\neg Q \rightarrow \neg P$ is true

Ex: Give indirect proof of the theorem "if 3n+ 2 is odd, then n is odd"
First, you need to change the theorem to become: "if n is even, then 3n+2 is even"

1. Assume that n is even so n =2K
2. 3n+ 2 = 3(2K) + 2 = 6K + 2 = 2(3K + 1) so it is even
3. If n is even then 3n + 2 is even,
   So if 3n+2 is odd then n is odd

3. Prove by Contradiction
Ex: proof by contradiction that " **if** n is an odd integer, then n^2 is an odd integer"
1. assume n is even but n^2 is an odd integer
2. n = 2K
3. $n^2 = 4K^2 = 2(2K^2)$ it's even
4. n^2 can't be odd and even in the same time.
   So by **<u>contradiction</u>** if n is even then n^2 is even.
   So if n is odd then n^2 is odd.

4. Proof by cases
Ex1: Use proof by cases to show that: |xy|=|x||y|
Case P1: $x>=0 \wedge x>=0$
Case P2: $x>=0 \wedge x<0$
Case P3: $x<0 \wedge x>=0$
Case P4: $x<0 \wedge x<0$
Case P: |x||y|
We need to show that
$P1 \rightarrow P \wedge P2 \rightarrow P \wedge P3 \rightarrow P \wedge P4 \rightarrow P$
$T \wedge T \wedge T \wedge T = T$
Ex2: Use proof by cases to show that: if n is even or odd integer then 2n+3 is odd.

Chapter 2

1 Sets

- **Def 1:** A set is an unordered collection of objects
- **Def 2:** The object in a set are also called the elements or members

- **Def 3:**

N = { 1,2,3 ….} the set of **natural numbers**
Z= { ….. , -2 , -1 , 0 , 1 ,2 , ….} the set of **integers.**
Z^+ = { 1, 2 ,3 …..} the set of **positive integers**
Q = {p/q $p \in Z$, $q \in Z$, $q \neq 0$} set of **rational numbers**
R, the set of real numbers

- **Def 4: Two sets are *equal* if and only if they have the same elements.**

Ex: { 1,3,5} and { 5, 1,3} are equal .
{5,1,3} and { 5,5,5,5,1,1,3,3} are equal

- **Def 5: Empty Set(Null set)** is a set with no elements. **Ex:** { }
- **Def 6: Singleton set** is the set with one element. **Ex:** { ∅} , {1} ,{A}
- **Def 7: Finite set** is the set with limited number of elements
  Infinite set is the set with unlimited number of elements

- **Def 8: Set cardinality (|S|)** is the number of elements in a set.

Ex: 1. S= { 1, 2, 3,-5 ,0} . |S| = 5
2. |∅| = 0

Set can be described by:

A. Listing all of it's members

Ex: describe the set of positive odd numbers less than 10.
O = { 1, 3, 5, 7, 9}

B. Set Builder Notation

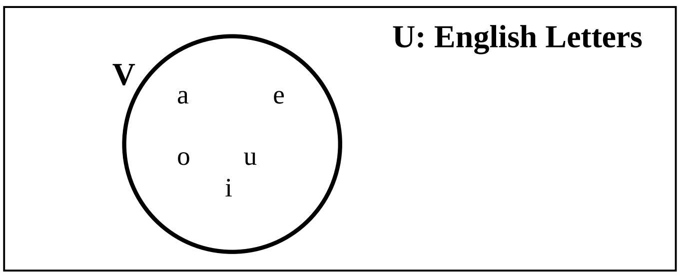
Ex: describe the set of odd numbers less that 10 using set builder notation.

O = { x | x is an odd positive number less than 10}

C. Venn diagram

Def: universal set *U* is the set that contains all objects under consideration.

Ex: V = describes the set of **Vowels** using Venn diagram.



Subsets ($A \subseteq B$)

- The set **A** is a subset of the set **B** if and only if every element of A is also an element of B. **($A \subseteq B$)**
- **($A \subseteq B$)** is true, if and only if the quantification $\forall x(x\in A \rightarrow x\in B)$ is true

- **Thm: for any set S**
  - * ($\varnothing \subseteq S$) **: { } is a subset of any set.**
  - * ($S \subseteq S$) **: any set is a subset of itself.**

- If **($A \subseteq B$)** is true and **($B \subseteq A$)** is true then **A = B.** $\forall x(x\in A \leftrightarrow x\in B)$ is true

Proper subset ($A \subset B$)

- The set **A** is a proper subset of the set **B** if and only if every element of A is also an element of B, but $A \neq B$. **($A \subset B$)**
- **($A \subset B$)** is true, if and only if the quantification $\forall x((x\in A \rightarrow x\in B) \wedge A \neq B)$ is true

Ex: $S = \{ \varnothing, 1, 2, 3, 4, 5, \{1\} \}$

$1\in S$	$6\notin S$	$\{1\} \subseteq S$	$\{1\} \subset S$
$S \subseteq S$	$S \not\subset S$	$\varnothing \subseteq S$	$\varnothing \in S$
$\{\varnothing\} \subseteq S$	$\{1\} \in S$	$\{\{1\}\} \subseteq S$	$\{1,2,3\} \subseteq S$

Power set P(S)

- Given a set S, the power set of S is the set of all subsets of the set S. the power set is denoted by ***P(S)***

Ex: what is the power set of the set {0, 1, 2}
P(S)= $\{ \varnothing , \{ 0 , 1, 2\} , \{0\} , \{1 \} , \{2 \} , \{0,1\} , \{0,2\} ,\{1 ,2\} \}$

Ex: $P(\{\varnothing\}) = \{ \varnothing , \{\varnothing\}\}$
$P(\varnothing) = \{\varnothing\}$

- If a set has n elements, then its power set has 2^n elements.

Ex: S = {0, 1, 2} then number of subsets is $2^3 = 8$

Cartesian Products

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

Ex: A = {1, 2} B= {a, b, c}
$A \times B = \{ (1,a), (1,b) ,(1,c) , (2,a), (2,b) ,(2,c)\}$
$B \times A = \{ (a,1), (a,2), (b,1), (b,2), (c,1), (c,2)\}$

- $A \times B \neq B \times A$
- ***Relation*** from the set A to the set B is a subset from $A \times B$

2 Set Operations

1. Union (A $\cup$ B): { x| x$\in$ A $\vee$ x$\in$B}

2. Intersection (A $\cap$ B): { x| x$\in$ A $\wedge$ x$\in$B}

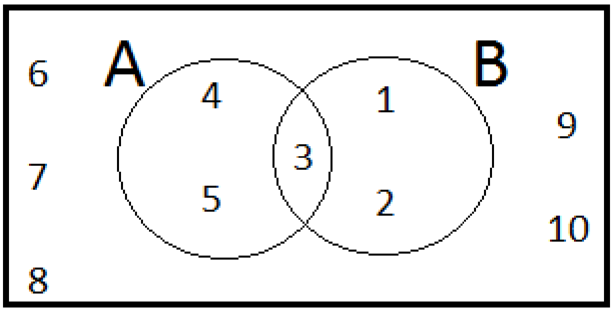
3. Difference (A –B) : { x| x$\in$ A $\wedge$ x$\notin$B}

4. Complement $\overline{A}$: { x| x$\notin$ A }

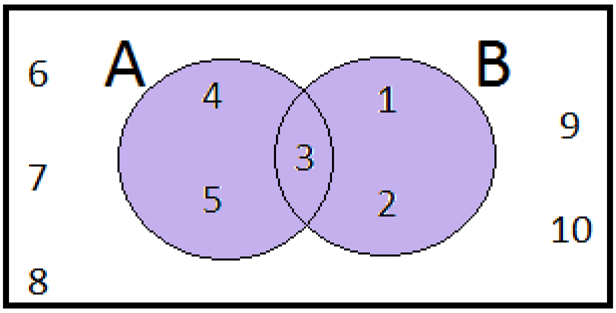
5. Symmetric Difference (A Δ B) : { x | x $\in$ A $\cup$ B $\wedge$ x $\notin$ A $\cap$ B}

Example:

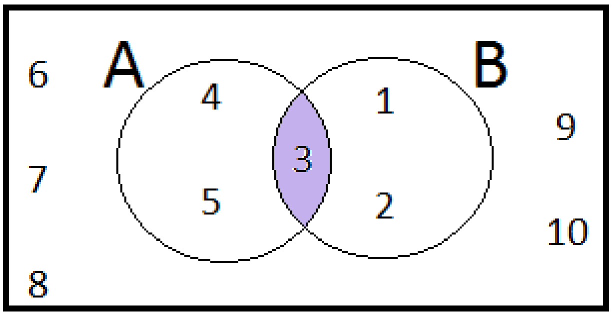
A = {3, 4, 5} B={1, 2, 3} U={1, 2, 3, … , 10}



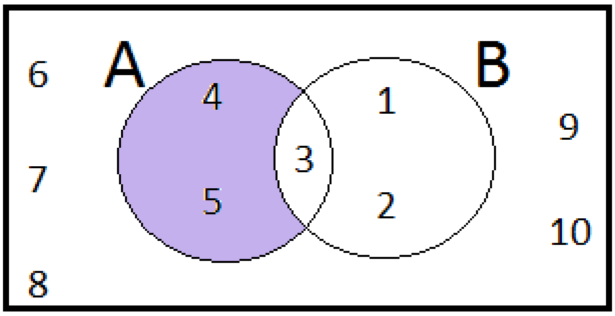
A $\cup$ B = {1, 2, 3, 4, 5}



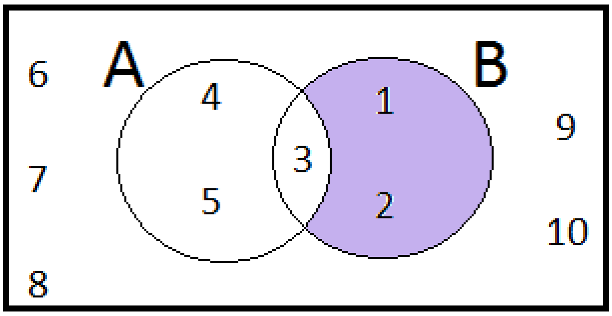
A $\cap$ B = {3}



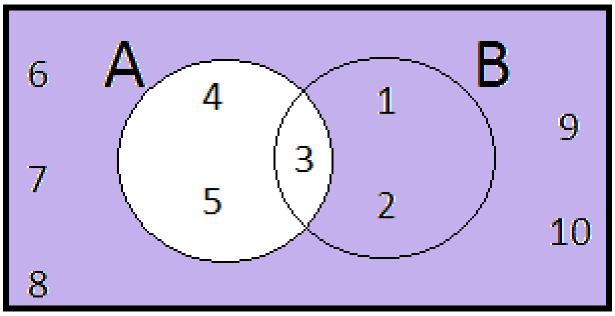
A – B = {4, 5}



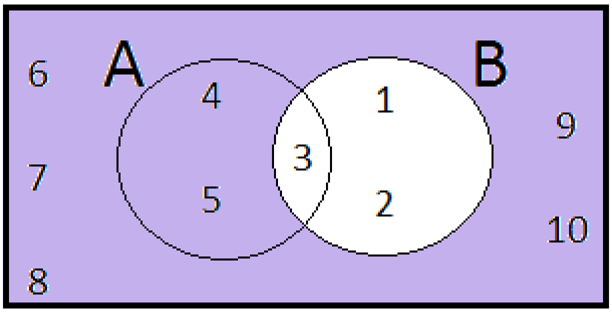
B – A = {1, 2}



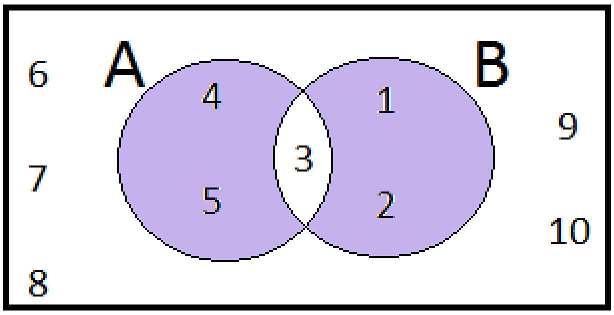
$\overline{\textbf{A}}$ **= { 1, 2, 6,7, 8, 9, 10}**



$\overline{\textbf{B}}$ **= { 4, 5, 6, 7, 8, 9, 10}**



$A \Delta B = \{1, 2, 4, 5\}$

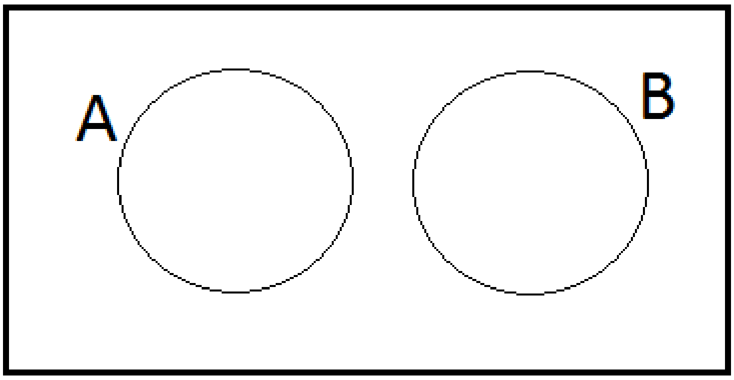


Def: **a. A and B are *disjoint* if and only if $A \cap B = \varnothing$**

Example : $A=\{x \mid x \text{ is an even number} \in Z\}$

$B=\{ x \mid x \text{ is an odd number} \in Z\}$

$A \cap B = \varnothing$



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

Example:

$A = \{3, 4, 5\}$ $B=\{1, 2, 3\}$ $A \cup B = \{ 1, 2, 3, 4, 5\}$ $A \cap B = \{3\}$

$| A \cup B | = 3 + 3 - 1 = 5$

c. $| A \cup B\cup C| = |A| + |B| + |C| - | A \cap B|- | A \cap C|- | B \cap C|- | A \cap B\cap C |$

* Computer representation of sets

Let U = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
What is the bit string that represents the set of odd integers in U?

Sol:

U=	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
S=	**1**		**3**		**5**		**7**		**9**	
	1	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**

Meaning of bit string:
1 → $x \in S$
0 → $x \notin S$

Set Operations:
Let U = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
Let A = 1 0 1 0 1 0 1 0 1 0 & B = 1 1 1 0 1 0 1 0 1 1
This means: A= {1,3,5,7,9} and B={1,2,3,5,7,9,10}

1 0 1 0 1 0 1 0 1 0 $\vee$ 1 1 1 0 1 0 1 0 1 1 = 1 1 1 0 1 0 1 0 1 1 = $A \cup B$

1 0 1 0 1 0 1 0 1 0 $\wedge$ 1 1 1 0 1 0 1 0 1 1 = 1 0 1 0 1 0 1 0 1 0 = $A \cap B$

1 1 1 0 1 0 1 0 1 1 $\wedge$ ~(1 0 1 0 1 0 1 0 1 0) = 0 1 0 0 0 0 0 0 0 1 = B – A

~(1 0 1 0 1 0 1 0 1 0) = 0 1 0 1 0 1 0 1 0 1 = $\overline{A}$

* Proving techniques

A. Set identities

Equivalence rule	Name
$A \cup \varnothing \equiv A$ $A \cap U \equiv A$	**Identity**
$A \cap \varnothing \equiv \varnothing$ $A \cup U \equiv U$	**Domination**
$A \cup A \equiv A$ $A \cap A \equiv A$	**Idempotent**
$(\overline{\overline{A}}) = A$	**Complementation law**
$A \cup B \equiv B \cup A$ $A \cap B \equiv B \cap A$	**Commutative**
$(A \cap B) \cap C \equiv A \cap (B \cap C)$ $(A \cup B) \cup C \equiv A \cup (B \cup C)$	**Associative**
$A \cap (B \cup C) \equiv (A \cap B) \cup (A \cap C)$ $A \cup (B \cap C) \equiv (A \cup B) \cap (A \cup C)$	**Distributive**
$\overline{A \cup B} \equiv \overline{A} \cap \overline{B}$ $\overline{A \cap B} \equiv \overline{A} \cup \overline{B}$	**Demorgan's**
$A \cup (A \cap B) \equiv A$ $A \cap (A \cup B) \equiv A$	**Absorption**
$A \cap \overline{A} \equiv \varnothing$ $A \cup \overline{A} \equiv U$	**Complement law**

Ex: prove that $\overline{A \cup (B \cap C)} \equiv (\overline{C} \cup \overline{B}) \cap \overline{A}$ **using set identities:**

$$\overline{A} \cap \overline{(B \cap C)} \equiv \overline{A} \cap (\overline{B} \cup \overline{C})$$

$$\equiv (\overline{B} \cup \overline{C}) \cap \overline{A}$$

$$\equiv (\overline{C} \cup \overline{B}) \cap \overline{A}$$

B. Set builder notation

Ex: use set builder notation and logical equivalence to show that

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

Sol:

$\overline{A \cap B} \equiv \{ x | x \notin A \cap B\}$

$\equiv \{ x | \neg(x \in A \cap B)\}$

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

$= \{ x | \quad x \notin A \vee x \notin B\}$

$= \{ x | \quad x \in \overline{A} \vee x \in \overline{B}\}$

$= \{ x | \quad x \in (\overline{A} \cup \overline{B})\} \quad = \quad (\overline{A} \cup \overline{B})$

Exercise: Prove the following using set builder notation :

$A - B \equiv A \cap \overline{B}$

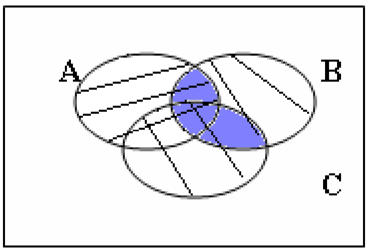
C. Membership table

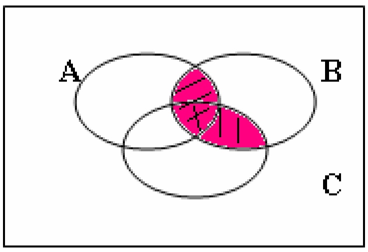
Ex: prove that A ∩ (B ∪ C) ≡ (A ∩ B) ∪ (A ∩ C) for all sets A, B, and C

A	B	C	B ∪ C	A ∩ (B ∪ C)	A ∩ B	A ∩ C	(A ∩ B) ∪ (A ∩ C)
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

D. Venn Diagrams

B ∩ (A ∪ C) ≡ (B ∩ A) ∪ (B ∩ C)





2. Functions

Def 1: Let A and B be sets, a function from A to B (f: A → B) is an assignment of exactly one elements of B to each element of A. where f(a) = b , and a ∈ A, b∈B.

Def 2: if f is a function from A to B:

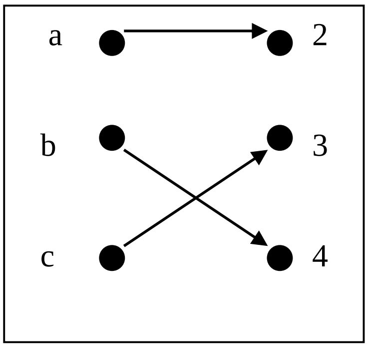
- **We say that A is the <u>domain</u> of f and B is the <u>codomain</u> of f.**
- **If $f(a)$ = b then a is the <u>pre-image</u> of b, and b is <u>image</u> of a.**
- **The <u>range</u> of f is the set of all images of elements of A.**
- **if f is a function from A to B, we say that A <u>maps</u> B.**

Ex: Let f: Z → Z such that f(x) = x^2 . the domain and the codomain is all integers. The range of f is the set positive integers Z^+

Ex: Let f: Z → Z such that f(x) = $x^{1/2}$. the domain and the codomain is all integers. This is **<u>not a function</u>** since negative values have no images and non perfect square have no integer images. But if f becomes from f:Z^+ → R^+, it becomes a function.

A function can be specified in different ways:

a. **Formula** ex: f(x) = x +1
b. **Graph** ex1: function f: A → B. A = { a , b , c} and B = { 2 , 3, 4}
f(a) = 2, f(b) = 4 f(c) = 3



Def 3: Let $f1$ and $f2$ be functions from A to R (i.e. *real valued functions*), then:

- **$f1$ + $f2$, and $f1\,f2$ are also functions**
- **($f1$ + $f2$)(x) = $f1$(x) + $f2$(x)**
- **($f1\,f2$)(x) = $f1$(x) $f2$(x)**

Ex: Let **$f1$** and **f2** be functions from **R** to **R** such that **$f1(x)$ = x^2** and **$f2(x)$ = x –x^2** what are the functions **$f1$ + $f2$** and **$f1\,f2$** for x =100**?**

($f1$ + $f2$)(x) = $f1$(x) + $f2$(x) = x^2 + (x –x^2) = x → ($f1$ + $f2$)(100)=100.
($f1\,f2$)(x) = $f1$(x) $f2$(x) = x^2 * (x –x^2) = x^3 – x^4 → ($f1\,f2$)(100) = 100^3-100^4.

Def 4: Identity function on A is the function ι_A: A → A , where f(x) = x
Ex: f: Z→Z , $f(x)$=x is an identity function

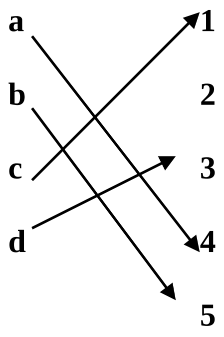
Functions Types

A. One –To - One (injective)

Def : A function *f* is One- to –One if and only if *f*(x) = *f*(y) implies that x =y for all x and y in the domain of *f*.

$\forall x \forall y\ (f(x) = f(y) \rightarrow x = y)$ **or** $\forall x \forall y\ (x \neq y \rightarrow f(x) \neq f(y))$

Ex: The function *f* from {a, b, c, d} to {1, 2, 3, 4, 5} with *f*(a) = 4, *f*(b) =5 , *f*(c) = 1, and *f*(d)=3 is **one - to –one.**



Ex: Determine whether the function $f: Z \rightarrow \mathbf{Z}$ such that $f(x) = x^2$ is one-to-one or not.

Sol. : The function $f(x) = x^2$ is **not one-to-one** . Because $f(1) = f(-1) = 1$ but $1 \neq -1$

Def: a function whose domain and codomain are subset of the set of real numbers is: (they are always one-to-one)

- **Strictly increasing if *f*(x) < *f*(y) whenever x < y and x, y are in the domain of *f*. example: *f*(x)= x + 2, *f*: Z → Z**

  $\forall x \forall y\ (x < y \rightarrow f(x) < f(y))$

- **Strictly Decreasing if *f*(x) > *f*(y) whenever x < y and x, y are in the domain of *f*. example: *f*(x)= 2 - x, *f*: Z → Z**

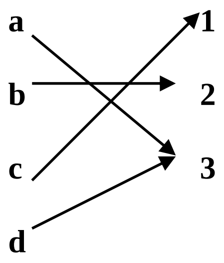
  $\forall x \forall y\ (x < y \rightarrow f(x) > f(y))$

B. Onto (Surjective)

Def: A function from A to B is onto if and only if every element b$\in$B there is an elements a $\in$ A with *f*(a) = b. (codomain=range)

$\forall y \exists x\ (f(x) = y)$

Ex: let the function *f* from {a, b, c, d} to { 1, 2 ,3} defined by *f*(a) = 3, *f*(b)= 2, *f*(c)=1, *f*(d) = 3, is *f* onto?



Onto function

Ex: is the function $f(x) = x^2$ from Z → Z Onto function.

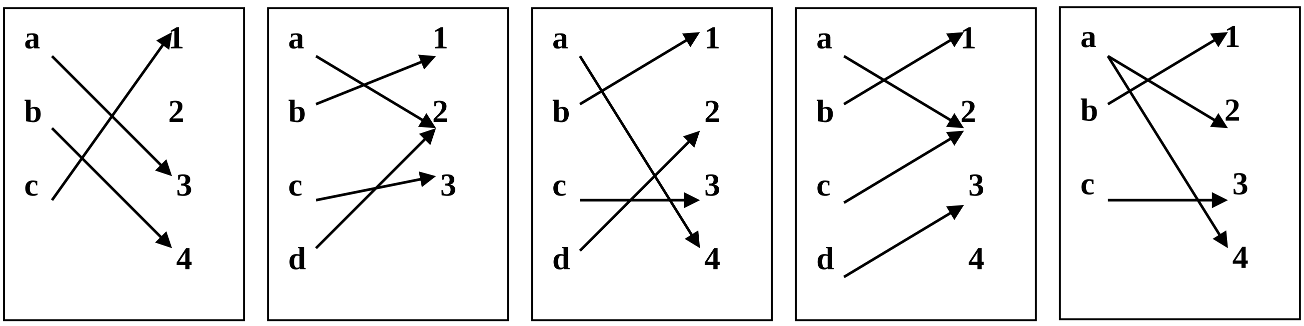
Sol.: It is **not onto** function, since there is no integer x such that $f(x) = -1$

C. One- to –one correspondence (bijective)

A function is **bijective** if and only if it is both one-to-one and onto.

Ex: identity function *f*(x) = x is bijective

Ex:



One-to-one
Not Onto

Onto
Not One-to-one

One-to-one
Onto
Bijective

Not Onto
Not one-to-one

Not a function

INVERSE AND COMPOSITE

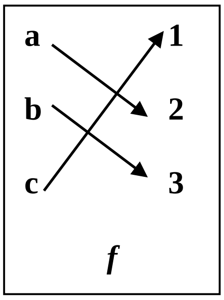
1. Inverse

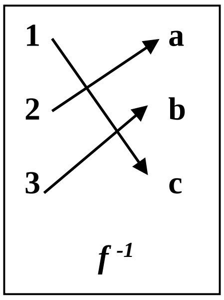
Def: let *f*: A → B be bijective from A to B. the inverse function of f is f^{-1}: B → A, where *f*(a) = b and f^{-1}(b) = a

- **Invertible function:** **bijective function is also called Invertible since we can define an inverse.**
- **Not Invertible function:** **if it is not bijective since we can't define an inverse.**
- $(f^{-1})^{-1} = f$

Ex: let *f* be the function from {a, b, c} to {1, 2, 3} such that *f(a)*=2, *f(b)*=3, *f(c)*=1, is *f* invertible and what is its inverse if it is?

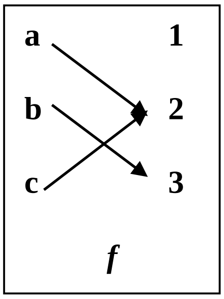
Sol.: *f* is invertible because it 's one-to-one correspondence
$f^{-1}(2)$=a, $f^{-1}(3)$=b, $f^{-1}(1)$=c

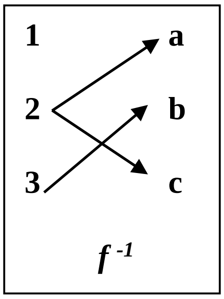




Ex: let *f* be the function from {a, b, c} to {1, 2, 3} such that *f(a)*=2, *f(b)*=3, *f(c)*=2, is *f* invertible and what is its inverse if it is?

Sol.: *f* is not invertible because it 's not one-to-one correspondence, since it is not one-to-one nor onto. But, if f^{-1} is constructed, the result **is not a function.**





Ex: Let *f* be the function from **Z** to **Z** with *f*(x) = x + 1, is *f* invertible?

Sol.: It is invertible and the inverse is $f^{-1}(y) = y - 1$

2. Composition

- **The composition of the functions f: A → B and g: B → C is denoted by :**
  (g o f)(a) = g(f(a))
- **if A ≠ C, then (f o g)(a) cant be calculated**
- **if f(a) =b . $(f^{-1} \circ f)(a) = (f^{-1}(f(a)) = f^{-1}(b) = a$ t_A**
- **if f(a) =b . $(f \circ f^{-1})(b) = f(f^{-1}(b)) = f(a) = b$ t_B**

Ex: Let $f(x) = 2x + 3$ and $g(x) = 3x + 2$ what is the composition of f and g if they both are from R to R?

$(f \circ g)(x) = f(g(x)) = f(3x + 2) = 6x + 7$

$(g \circ f)(x) = g(f(x)) = g(2x + 3) = 6x + 11$

- **$\therefore (f \circ g)(x) \neq (g \circ f)(x)$**

Important Functions

- ***Floor* function $\lfloor x \rfloor$: the floor of real number x is the largest integer that is less than or equal to x.**
- ***Ceiling* function $\lceil x \rceil$: the ceiling of real number x is the smallest integer that is greater than or equal to x.**

Ex: what is the value of the following?

$\lfloor 1/2 \rfloor = 0$ $\lceil 1/2 \rceil = 1$

$\lfloor -1/2 \rfloor = -1$ $\lceil -1/2 \rceil = 0$

$\lfloor 3.1 \rfloor = 3$ $\lceil 3.1 \rceil = 4$

$\lfloor 7 \rfloor = 7$ $\lceil 7 \rceil = 7$

Table: Properties of ceiling and flooring functions

$x-1 < \lfloor x \rfloor \leq x \leq \lceil x \rceil < x + 1$
$\lfloor -x \rfloor = -\lceil x \rceil$ $\lceil -x \rceil = -\lfloor x \rfloor$
$\lceil x + n \rceil = \lceil x \rceil + n$ **where n is an integer** $\lfloor x + n \rfloor = \lfloor x \rfloor + n$

3. Sequences and summation

1. Sequence

Def 1:

- Sequence is a function from a subset of the set of integers to a set S.
- we use the notation a_n to denote the image of the integers n.W
- We call the term a_n a *term* of the sequence.
- The list of terms is beginning with a_1: a_1, a_2, ….., a_n

Ex: Consider the sequence $\{a_n\}$, where $a_n = 1/n$
$a_1 = 1/1$, $a_2 = 1/2$, $a_3 = 1/3$ ………. Etc

Def 2: Sequences are two types

a. **Geometric progression**:

It is a sequence of form $a_0\ r^0,\ a_0\ r^1, a_0\ r^2,\ a_0\ r^3$ ………,$a_0\ r^n$, where n>=0
where the *initial term* is $\mathbf{a_0}$ and the *common ratio* **r** are real numbers.
Or
It is a sequence of form $a_1\ r^0,\ a_1\ r^1, a_1\ r^2,\ a_1\ r^3$ ………,$a_1\ r^{n-1}$, where n>=1
where the *initial term* is $\mathbf{a_1}$ and the *common ratio* **r** are real numbers.

b. **Arithmetic progression**:

It is a sequence of form $a_0 + 0d, a_0 + 1d, a_0 + 2d$,………, $a_0 + nd$, where n>=0
where the *initial term* $\mathbf{a_0}$ and the *common difference* **d** are real numbers.
Or
It is a sequence of form $a_1 + 0d, a_1 + 1d, a_1 + 2d$,………, $a_1 + n(d-1)$, where n>=1
where the *initial term* $\mathbf{a_1}$ and the *common difference* **d** are real numbers.

Ex: the sequences: $\{b_n\}$ with $b_n = (-1)^n$, $\{C_n\}$with $C_n = 2.5^n$ where n>=0 are **geometric progression** sequences.

$\{b_n\} = \{-1, 1, -1, 1,\ldots\}$ initial term = -1, common ratio= -1
$\{C_n\} = \{10,50,250,1250,\ldots\}$ initial term = 10, common ratio= 5

Ex: The sequence $\{S_n\}$ with Sn = -1 + 4n , where n>=0 is **arithmetic sequence**
Where {Sn} = {-1, 3, 7, 11,………}
Initial term = -1, with common Difference= 4

Strings

Def:

- The finite sequences are called strings
- The length of the string is the number of terms in this string.
- The empty string, denoted by λ, and it's the string with no terms.
- The empty string is with length zero.

Ex: the string "abcd" is with length 4.

Special, Integer Sequences

Ex: Find a formula for the following sequences:

A. A. 1, ½, ¼, 1/8 , 1/16

Sol: $a_n = 1/ 2^{n-1}$ it a geometric progression with initial term = 1 and common ration= ½

B. 5, 11, 17, 23, 29 …

Sol: $a_n = 6n – 1$ is arithmetic progression with a= 5, and d= 6

<u>Useful sequences</u>

Nth term	First 5 terms
n^2	1, 4, 9, 16, 25,……..
2^n	2, 4, 8, 16, 32,………
n!	1, 2, 6, 24, 120, …….

2. Summation

- The summation notation is: $\sum_{j=m}^{n} a_j$ or $\sum_{j=m}^{n} a_j$ to represent $a_m + a_{m+1}, \ldots + a_n$

Variable **j** is called the index of summation. M is lower limit, and n is the upper limit

- $\sum_{j=m}^{n} a_j = \sum_{i=m}^{n} a_i = \sum_{K=m}^{n} a_k$

Ex:

$$\begin{aligned}\sum_{i=2}^{4}(i^2+1) &= (2^2+1)+(3^2+1)+(4^2+1)\\ &= (4+1)+(9+1)+(16+1)\\ &= 5+10+17\\ &= 32\end{aligned}$$

Ex:

$$\begin{aligned}\sum_{i=1}^{4}\sum_{j=1}^{3} ij &= \sum_{i=1}^{4}\left(\sum_{j=1}^{3} ij\right) = \sum_{i=1}^{4} i\left(\sum_{j=1}^{3} j\right) = \sum_{i=1}^{4} i(1+2+3)\\ &= \sum_{i=1}^{4} 6i = 6\sum_{i=1}^{4} i = 6(1+2+3+4)\\ &= 6\times 10 = 60\end{aligned}$$

Ex: $\sum_{j=1}^{5} j^2 = \sum_{k=0}^{4}(K+1)^2 = \sum_{L=2}^{6}(L-1)^2 =$ = 1 + 4+ 9 + 16 + 25 = 55

Useful summation formula

1 $\quad \sum_{k=0}^{n} ar^k = a(r^{n+1}-1)/(r-1), r \neq 1$

2 $\quad \sum_{k=1}^{n} k = n(n+1)/2$

3 $\quad \sum_{k=1}^{n} k^2 = n(n+1)(2n+1)/6$

4 $\quad \sum_{k=1}^{n} k^3 = n^2(n+1)^2/4$

5 $\quad \sum_{K=o}^{\infty} x^k = 1/(1-x), \ |x| < 1$

6 $\quad \sum_{K=1}^{\infty} kx^{k-1} = 1/(1-x)^2, \ |x| < 1$

Ex: find the value of

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

Sol:

$$\sum_{k=1}^{100} k^2 = \left(\sum_{k=1}^{49} k^2\right) + \sum_{k=50}^{100} k^2$$

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

Ex: find the value of

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

Sol:

$$\sum_{k=2}^{100} k^2 = \sum_{k=1}^{100} k^2 - (1)^2$$

Ex: find the value of

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

Sol:

$$\sum_{k=-2}^{100} k^2 = \sum_{k=1}^{100} k^2 + (-2)^2 + (-1)^2 + (0)^2$$

Ex: Given that $\sum_{k=1}^{100} k^2 = 338{,}350$ find the value of :

a.

$$\sum_{k=1}^{99} k^2$$

b.

$$\sum_{k=1}^{101} k^2$$

Sol:

a.

$$\sum_{k=1}^{99} k^2 = \sum_{k=1}^{100} k^2 - (100)^2 = 338{,}350 - 10{,}000 = 328{,}350$$

b.

$$\sum_{k=1}^{101} k^2 = \sum_{k=1}^{100} k^2 + (101)^2 = 338{,}350 + 10{,}201 = 348{,}551$$

Module #3:
The Theory of Sets

Rosen 5th ed., §§1.6-1.7
~43 slides, ~2 lectures

Introduction to Set Theory (§1.6)

- A *set* is a new type of structure, representing an *unordered* collection (group, plurality) of zero or more *distinct* (different) objects.
- Set theory deals with operations between, relations among, and statements about sets.
- Sets are ubiquitous in computer software systems.
- *All* of mathematics can be defined in terms of some form of set theory (using predicate logic).

Naïve set theory

- **Basic premise:** Any collection or class of objects (*elements*) that we can *describe* (by any means whatsoever) constitutes a set.
- But, the resulting theory turns out to be *logically inconsistent*!
  - This means, there exist naïve set theory propositions p such that you can prove that both p and $\neg p$ follow logically from the axioms of the theory!
  - $\therefore$ The conjunction of the axioms is a contradiction!
  - This theory is fundamentally uninteresting, because any possible statement in it can be (very trivially) "proved" by contradiction!
- More sophisticated set theories fix this problem.

Basic notations for sets

- For sets, we'll use variables S, T, U, …
- We can denote a set S in writing by listing all of its elements in curly braces:
  - $\{a, b, c\}$ is the set of whatever 3 objects are denoted by a, b, c.
- *Set builder notation*: For any proposition $P(x)$ over any universe of discourse, $\{x|P(x)\}$ is *the set of all x such that P(x)*.

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, 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.*
- **For example:** The set $\{1, 2, 3, 4\}$ =
  $\{x \mid x \text{ is an integer where } x>0 \text{ and } x<5\}$ =
  $\{x \mid x \text{ is a positive integer whose square is } >0 \text{ and } <25\}$

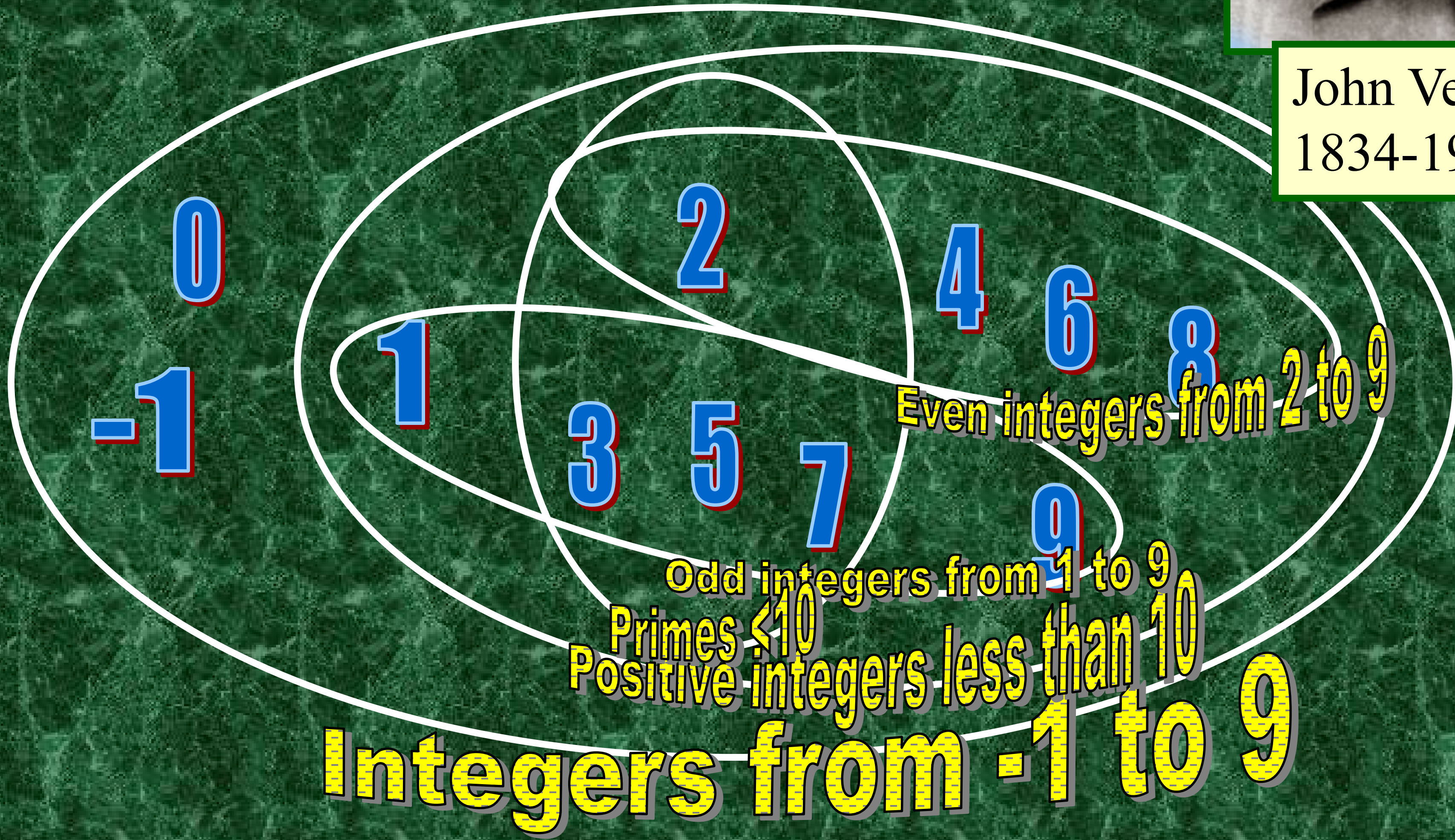
Infinite Sets

- Conceptually, sets may be *infinite* (*i.e.*, not *finite*, without end, unending).
- Symbols for some special infinite sets:
  N = {0, 1, 2, …} The **N**atural numbers.
  Z = {…, -2, -1, 0, 1, 2, …} The **Z**ntegers.
  R = The "**R**eal" numbers, such as 374.1828471929498181917281943125…
- "Blackboard Bold" or double-struck font ($\mathbb{N}$,$\mathbb{Z}$,$\mathbb{R}$) is also often used for these special number sets.
- Infinite sets come in different sizes!

More on this after module #4 (functions).

Venn Diagrams

John Venn
1834-1923



Basic Set Relations: Member of

- $x \in S$ ("x is in S") is the proposition that object x is an $\in$*lement* or *member* of set S.
  - *e.g.* $3 \in \mathbf{N}$, "a" $\in \{x \mid x$ 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

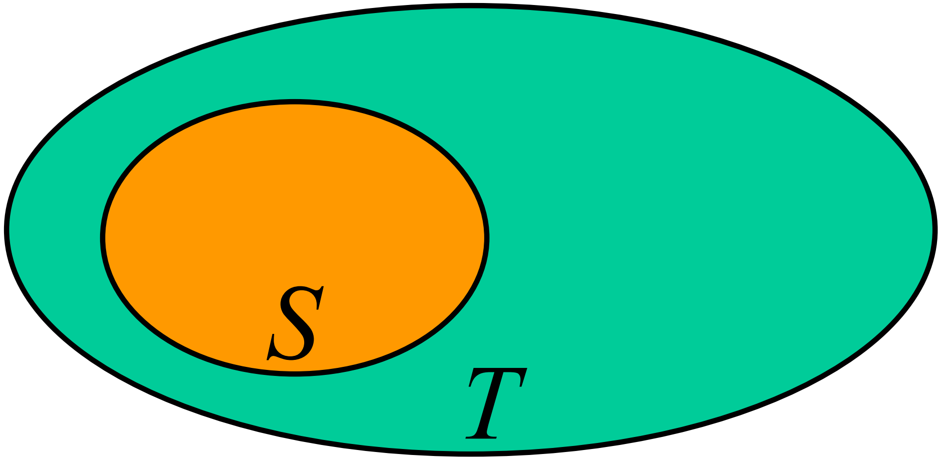
- $\varnothing$ ("null", "the empty set") is the unique set that contains no elements whatsoever.
- $\varnothing = \{\} = \{x|\mathbf{False}\}$
- No matter the domain of discourse, we have the axiom $\neg\exists x\colon x\in\varnothing$.

Subset and Superset Relations

- $S \subseteq T$ ("S is a subset of T") means that every element of S is also an element of T.
- $S \subseteq T \Leftrightarrow \forall x\, (x \in S \rightarrow x \in T)$
- $\emptyset \subseteq S$, $S \subseteq S$.
- $S \supseteq T$ ("S is a superset of T") means $T \subseteq S$.
- Note $S = T \Leftrightarrow S \subseteq T \wedge S \supseteq T$.
- $S \not\subseteq T$ 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 equivalent of $S \subset T$

Sets Are Objects, Too!

- The objects that are elements of a set may *themselves* be sets.
- *E.g.* let $S=\{x \mid x \subseteq \{1,2,3\}\}$
  then $S=\{\varnothing,$
  $\{1\}, \{2\}, \{3\},$
  $\{1,2\}, \{1,3\}, \{2,3\},$
  $\{1,2,3\}\}$
- Note that $1 \neq \{1\} \neq \{\{1\}\}$!!!!

Very Important!

Cardinality and Finiteness

- $|S|$ (read "the *cardinality* of S") is a measure of how many different elements S has.
- *E.g.*, $|\varnothing|=0$, $|\{1,2,3\}| = 3$, $|\{a,b\}| = 2$,
  $|\{\{1,2,3\},\{4,5\}\}| =$ __2__
- 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 R

The *Power Set* Operation

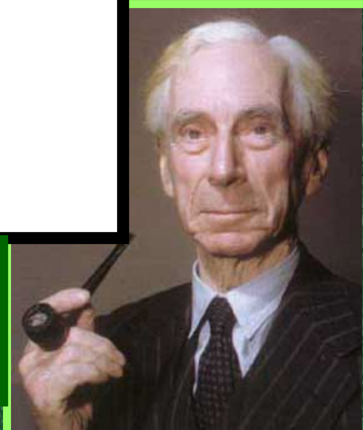
- The *power set* P(S) of a set S is the set of all subsets of S. P(S) :≡ $\{x \mid x \subseteq S\}$.
- *E.g.* P({a,b}) = {∅, {a}, {b}, {a,b}}.
- Sometimes P(S) is written $\mathbf{2}^S$.
  Note that for finite S, $|P(S)| = 2^{|S|}$.
- It turns out $\forall S{:}|P(S)|>|S|$, *e.g.* $|P(\mathbf{N})| > |\mathbf{N}|$.
  There are different sizes of infinite sets!

Review: Set Notations So Far

- Variable objects x, y, z; sets S, T, U.
- Literal set {a, b, c} and set-builder $\{x|P(x)\}$.
- $\in$ relational operator, and the empty set $\emptyset$.
- Set relations $=$, $\subseteq$, $\supseteq$, $\subset$, $\supset$, $\not\subset$, etc.
- Venn diagrams.
- Cardinality $|S|$ and infinite sets **N**, **Z**, **R**.
- Power sets P(S).

Naïve Set Theory is Inconsistent

- There are some naïve set *descriptions* that lead to pathological structures that are not *well-defined*.
  - (That do not have self-consistent properties.)
- These "sets" mathematically *cannot* exist.
- *E.g.* let $S = \{x \mid x \notin x\}$. Is $S \in S$?
- Therefore, consistent set theories must restrict the language that can be used to describe sets.
- For purposes of this class, don't worry about it!



Bertrand Russell
1872-1970

Ordered n-tuples

- These are like sets, except that duplicates matter, and the order makes a difference.
- For $n \in \mathbf{N}$, an *ordered n-tuple* or a *sequence* or *list of length n* is written $(a_1, a_2, \ldots, a_n)$. Its *first* element is a_1, *etc.*
- Note that $(1, 2) \neq (2, 1) \neq (2, 1, 1)$.
- Empty sequence, singlets, pairs, triples, quadruples, quin<u>tuples</u>, …, n-tuples.

Contrast with sets' {}

Cartesian Products of Sets

- For sets A, B, their *Cartesian product* $A \times B :\equiv \{(a, b) \mid a \in A \wedge b \in B\}$.
- *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.*, $\neg\forall AB: A \times B = B \times A$.
- Extends to $A_1 \times A_2 \times \ldots \times A_n$...

René Descartes
(1596-1650)

Review of §1.6

- Sets S, T, U… Special sets **N**, **Z**, **R**.
- Set notations $\{a,b,...\}$, $\{x|P(x)\}$…
- Set relation operators $x\in S$, $S\subseteq T$, $S\supseteq T$, $S=T$, $S\subset T$, $S\supset T$. (These form propositions.)
- Finite vs. infinite sets.
- Set operations $|S|$, $P(S)$, $S\times T$.
- Next up: §1.5: More set ops: $\cup$, $\cap$, $-$.

Start §1.7: The Union Operator

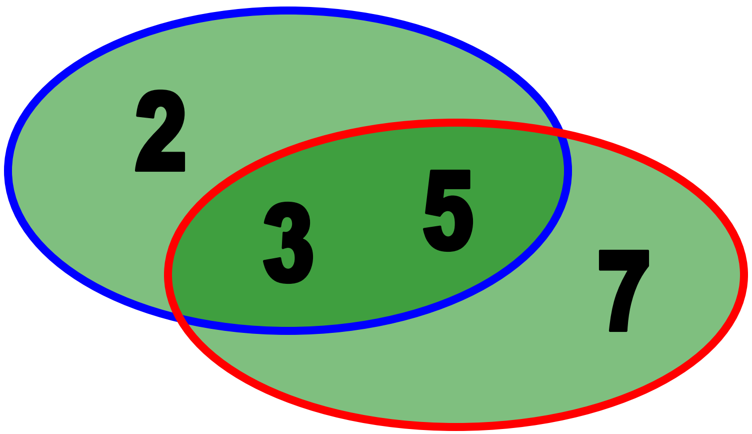
- For sets A, B, their *union* $A \cup B$ is the set containing all elements that are either in A, **or** ("$\vee$") in B (or, of course, in both).
- Formally, $\forall A, B: A \cup B = \{x \mid x \in A \vee x \in B\}$.
- Note that $A \cup B$ is a **superset** of both A and B (in fact, it is the smallest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

$$\forall A, B: (A \cup B \supseteq A) \wedge (A \cup B \supseteq B)$$

Union Examples

Required Form

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



Think "The **Uni**ted States of America includes every person who worked in any U.S. state last year." (This is how the IRS sees it...)

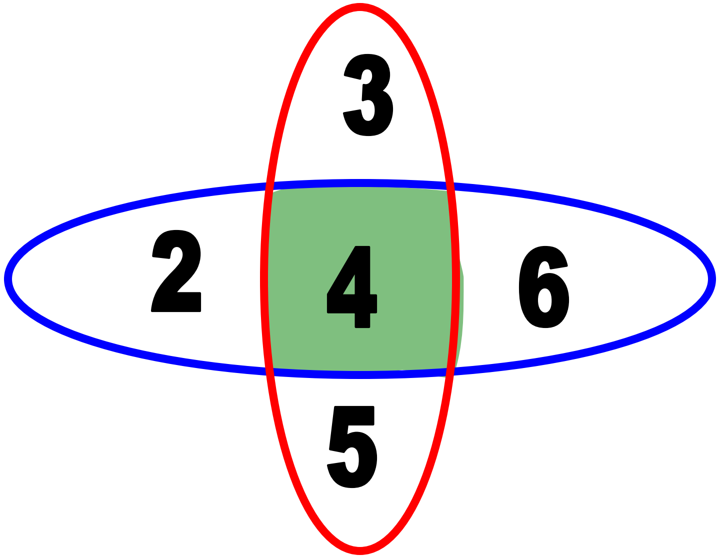
The Intersection Operator

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

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

Intersection Examples

- $\{a,b,c\} \cap \{2,3\} = \underline{\ \varnothing\ }$
- $\{2,4,6\} \cap \{3,4,5\} = \underline{\quad \{4\} \quad}$



Think "The **<u>intersection</u>** of University Ave. and W 13th 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 are on our class email list? Consider set $E = I \cup M$,

  $I = \{s \mid s \text{ turned in an information sheet}\}$

  $M = \{s \mid s \text{ sent the TAs their email address}\}$

- Some students did both!

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

Set Difference

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

$$A - B :\equiv \{x \mid x \in A \land x \notin B\}$$
$$= \{x \mid \neg(x \in A \rightarrow x \in B)\}$$

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

Set Difference Examples

- $\{1,\cancel{2},\cancel{3},4,\cancel{5},6\} - \{2,3,5,7,9,11\} =$

  $\underline{\quad\{1,4,6\}\quad}$

- $\mathbf{Z} - \mathbf{N} = \{\ldots, -1, 0, 1, 2, \ldots\} - \{0, 1, \ldots\}$

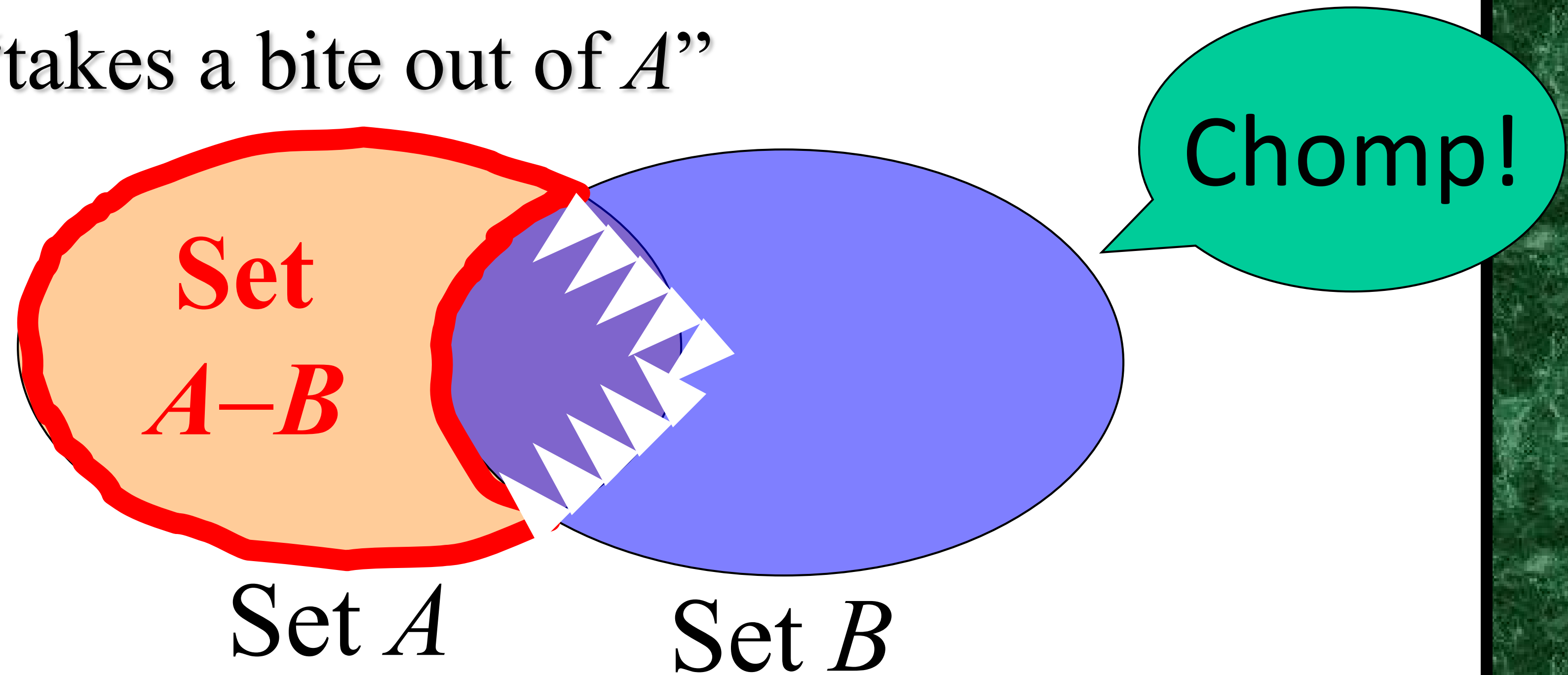
  $= \{x \mid x \text{ is an integer but not a nat. \#}\}$

  $= \{x \mid x \text{ is a negative integer}\}$

  $= \{\ldots, -3, -2, -1\}$

Set Difference - Venn Diagram

- $A-B$ is what's left after B "takes a bite out of A"



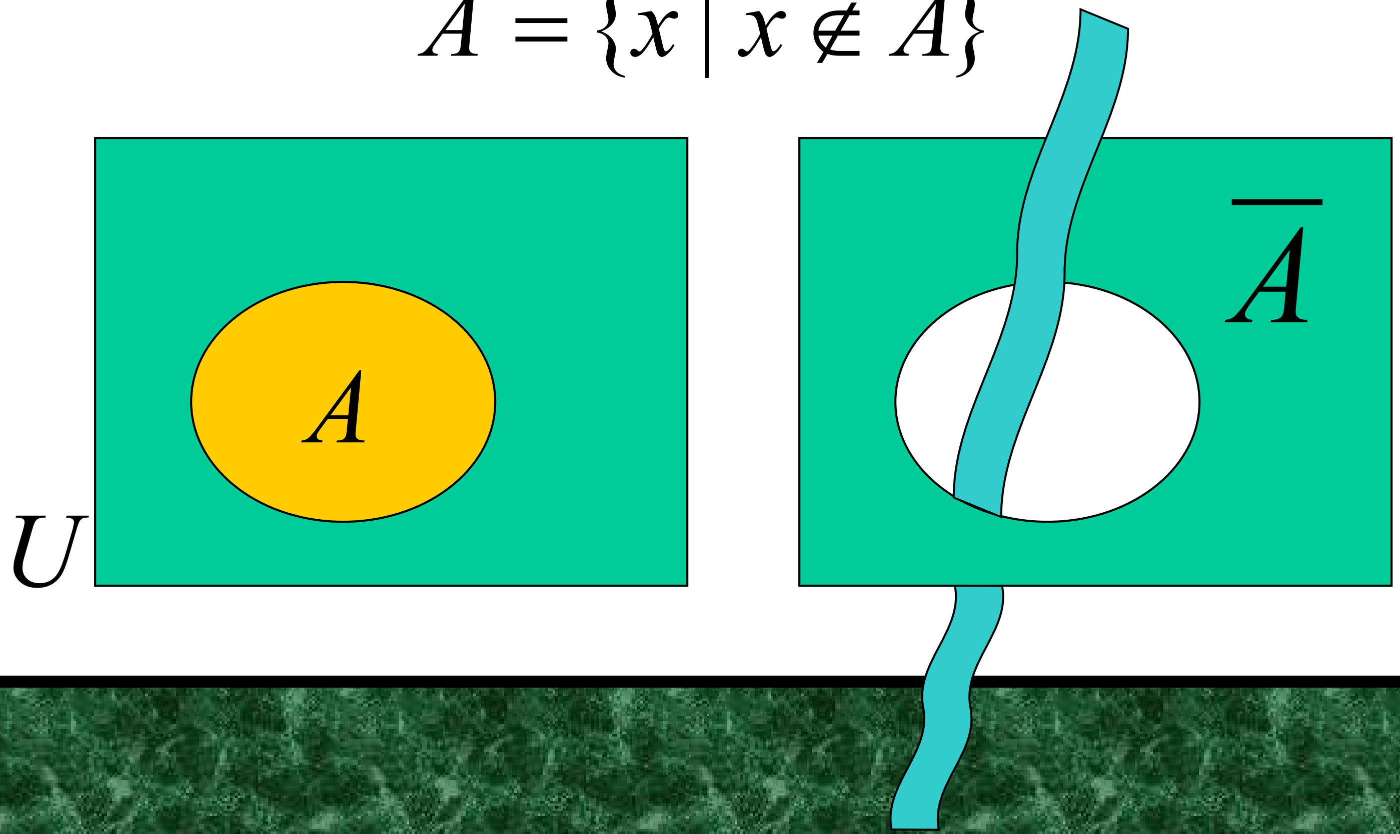
Set Complements

- The *universe of discourse* 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 $\overline{A}$, is the complement of A w.r.t. U, *i.e.*, it is $U-A$.
- *E.g.*, If $U=\mathbf{N}$, $\overline{\{3,5\}} = \{0,1,2,4,6,7,\ldots\}$

More on Set Complements

- An equivalent definition, when *U* is clear:

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



Set Identities

- Identity: $A \cup \varnothing = A = A \cap U$
- Domination: $A \cup U = U$, $A \cap \varnothing = \varnothing$
- Idempotent: $A \cup A = A = A \cap A$
- 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: $A \cup (B \cap C) = (A \cup B) \cap (\mathrm{A} \cup C)$,
  $A \cap (B \cup C) = (A \cap B) \cup (\mathrm{A} \cap C)$

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 Es are set expressions), here are three useful techniques:

1. Using Set Identities
2. Prove $E_1 \subseteq E_2$ and $E_2 \subseteq E_1$ separately.
3. Use set builder notation & Logical equivalences.
4. Use a *membership table*.
5. Venn Diagram

Method 1: Set Identities

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

$(A \cup B) - B = A \cup B \cap \overline{B}$ → Definition of –

$(A \cup B) \cap \overline{B} = A \cup \overline{B} \cap B \cup \overline{B}$ → Distributive Law

$A \cap \overline{B} \cup B \cap \overline{B} = A \cap B \cup \varnothing$ → Domination Law

$A \cap \overline{B}$ → Identity Law

$A - B$ → Definition of –

Method 2: 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: $x\in B$. Then $x\in A\cap B$, so $x\in(A\cap B)\cup(A\cap C)$.
    - Case 2: $x\in C$. 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 $(A\cap B)\cup(A\cap C) \subseteq A\cap(B\cup C)$. …

Method 3: Set Builder Notation

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

$(A\cup B)-B = \{x \mid x\in (A\cup B)-B\}$

$=\{x|x\in (A\cup B) \wedge x\notin B\}=\{x|x\in A \vee x\in B \wedge \sim x\in B\}$

$=\{x|x\in A\vee B\wedge\sim B\}=\{x|x\in(A\wedge\sim B)\vee(B\wedge\sim B)\}$

$=\{x|x\in(A\wedge\sim B)\vee F\}= \{x|x\in(A\wedge\sim B)\}$

$=\{x|x\in(A\cap\sim B)\}=\{x|x\in(A-B)\}= A-B$

Method 4: Membership Tables

- Just like truth tables for 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	$A \cup B$	$(A \cup B) - B$	$A - B$
0	0	0	0	0
0	1	1	0	0
1	0	1	1	1
1	1	1	0	0

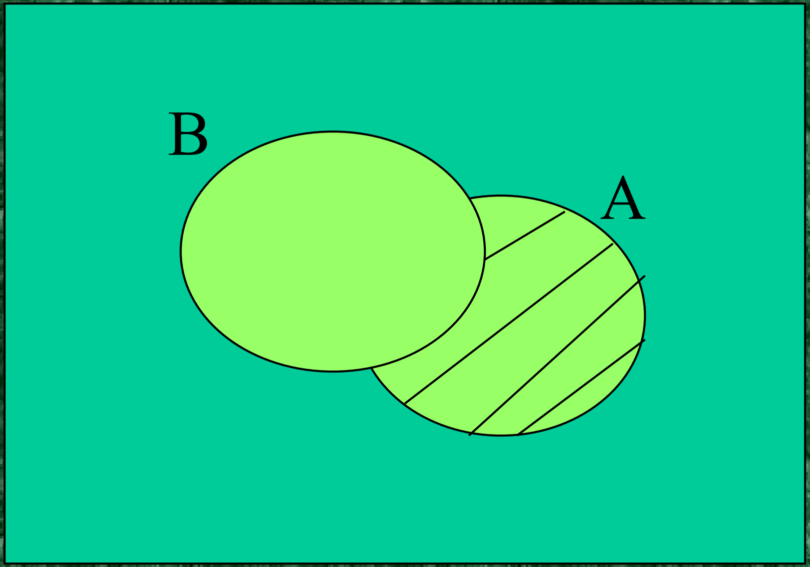
Membership Table Exercise

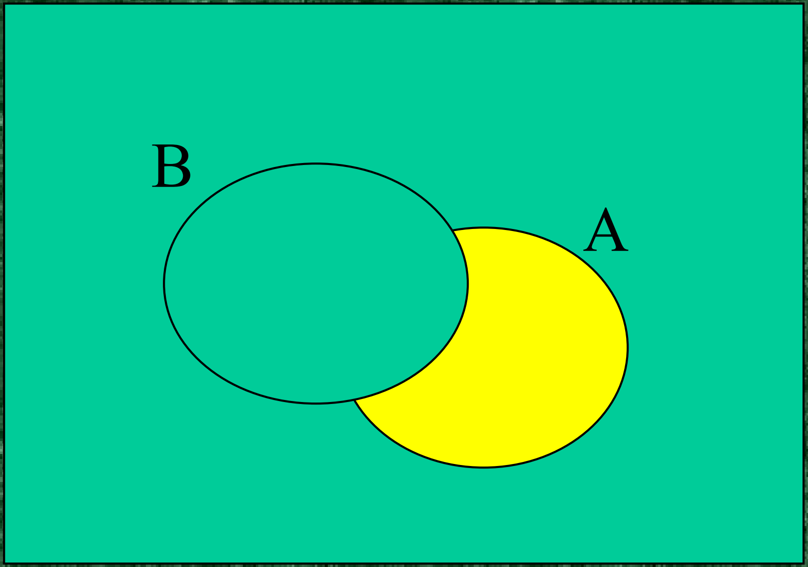
Prove $(A \cup B) - C = (A - C) \cup (B - C)$.

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

Method 5: Venn Diagram

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





Review of §1.6-1.7

- Sets S, T, U… Special sets $\mathbf{N}$, $\mathbf{Z}$, $\mathbf{R}$.
- Set notations $\{a,b,...\}$, $\{x|P(x)\}$…
- Relations $x\in S$, $S\subseteq T$, $S\supseteq T$, $S=T$, $S\subset T$, $S\supset T$.
- Operations $|S|$, $P(S)$, $\times$, $\cup$, $\cap$, $-$, $\overline{S}$
- Set equality proof techniques:
  - Mutual subsets.
  - Derivation using logical equivalences.

Generalized Unions & Intersections

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

Generalized Union

- Binary union operator: $A \cup B$
- n-ary union:
  $A \cup A_2 \cup \ldots \cup A_n :\equiv ((\ldots((A_1 \cup A_2) \cup \ldots) \cup A_n)$
  (grouping & order is irrelevant)
- "Big U" notation: $\bigcup_{i=1}^{n} A_i$
- Or for infinite sets of sets: $\bigcup_{A \in X} A$

Generalized Intersection

- Binary intersection operator: $A \cap B$
- n-ary intersection:
  $A_1 \cap A_2 \cap \ldots \cap A_n \equiv ((\ldots((A_1 \cap A_2) \cap \ldots) \cap A_n)$
  (grouping & order is irrelevant)
- "Big Arch" notation: $\bigcap_{i=1}^{n} A_i$
- Or for infinite sets of sets: $\bigcap_{A \in X} A$

Representations

- A frequent theme of this course will be methods of *representing* one discrete structure using another discrete structure of a different type.
- *E.g.*, one can represent natural numbers as
  - Sets: **0**:≡∅, **1**:≡{**0**}, **2**:≡{**0**,**1**}, **3**:≡{**0**,**1**,**2**}, …
  - Bit strings: **0**:≡`0`, **1**:≡`1`, **2**:≡`10`, **3**:≡`11`, **4**:≡`100`, …

Representing Sets with Bit Strings

For an enumerable u.d. U with ordering $x_1, x_2, \ldots$, represent a finite set $S \subseteq U$ as the finite bit string B=$b_1 b_2 \ldots b_n$ where $\forall i$: $x_i \in S \leftrightarrow (i<n \wedge b_i=1)$.

E.g. $U=\mathbf{N}$, $S=\{2,3,5,7,11\}$, B=001101010001.

In this representation, the set operators "$\cup$", "$\cap$", "$\overline{}$" are implemented directly by bitwise OR, AND, NOT!

Inductive Proofs

Mathematical Induction

- A powerful, rigorous technique for proving that a predicate $P(n)$ is true for *every* natural number n, no matter how large.
- Essentially a "domino effect" principle.
- Based on a predicate-logic inference rule:

$$\begin{array}{l} P(0) \\ \forall n{\geq}0\ (P(n) \rightarrow P(n+1)) \\ \hline \therefore \forall n{\geq}0\ P(n) \end{array}$$

"The First Principle of Mathematical Induction"

Validity of Induction

Proof that $\forall k \geq 0\ P(k)$ is a valid consequent:

Given any $k \geq 0$, $\forall n \geq 0\ (P(n) \rightarrow P(n+1))$ (antecedent 2) trivially implies $\forall n \geq 0\ (n < k) \rightarrow (P(n) \rightarrow P(n+1))$, or $(P(0) \rightarrow P(1)) \wedge (P(1) \rightarrow P(2)) \wedge \ldots \wedge (P(k-1) \rightarrow P(k))$. Repeatedly applying the hypothetical syllogism rule to adjacent implications k-1 times then gives $P(0) \rightarrow P(k)$; which with $P(0)$ (antecedent #1) and *modus ponens* gives $P(k)$. Thus $\forall k \geq 0\ P(k)$.

Outline of an Inductive Proof

- Want to prove $\forall n\, P(n)$…
- *Base case* (or *basis step*): Prove $P(0)$.
- *Inductive step*: Prove $\forall n\, P(n) \rightarrow P(n+1)$.
  - *E.g.* use a direct proof:
  - Let $n \in \mathbf{N}$, assume $P(n)$. (*inductive hypothesis*)
  - Under this assumption, prove $P(n+1)$.
- Inductive inference rule then gives $\forall n\, P(n)$.

Induction Example

- Prove that the sum of the first n odd positive integers is n^2. That is, prove:

$$\underbrace{\forall n \geq 1: \sum_{i=1}^{n} (2i-1) = n^2}_{P(n)}$$

- Proof by induction.
  - Base case: Let n=1. The sum of the first 1 odd positive integer is 1 which equals 1^2. (Cont…)

Example cont.

- Inductive step: Prove $\forall n \geq 1$: $P(n) \rightarrow P(n+1)$.
  - Let $n \geq 1$, assume $P(n)$, and prove $P(n+1)$.

$$\sum_{i=1}^{n+1}(2i-1) = \left(\sum_{i=1}^{n}(2i-1)\right) + (2(n+1)-1)$$

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

By inductive hypothesis P(n)

$$= (n+1)^2$$

Another Induction Example

- Prove that $\forall n>0$, $n<2^n$. Let $P(n)=(n<2^n)$
  - Base case: $P(1)=(1<2^1)=(1<2)=\mathbf{T}$.
  - Inductive step: For $n>0$, prove $P(n)\rightarrow P(n+1)$.
    - Assuming $n<2^n$, prove $n+1 < 2^{n+1}$.
    - Note $n + 1 < 2^n + 1$ (by inductive hypothesis)
      $< 2^n + 2^n$ (because $1<2=2\cdot 2^0\leq 2\cdot 2^{n-1}= 2^n$)
      $= 2^{n+1}$
    - So $n + 1 < 2^{n+1}$, and we're done. i.e. P(n+1) is true

Another Induction Example

Use mathematical Induction to prove that the sum of the first n odd positive integers is n^2.

<u>SOL:</u>

A. Basic Step: p(1) , the sum of the first odd positive integer which is 1 is 1^2 and equal to 1. So, p(1) is true.

B. Inductive step: Suppose that p(k) is true.

So, $\mathbf{1+3+5+\ldots.+(2K-1)=k^2}$

We must show that p(K+1) is true, assuming that p(K) is true

$\mathbf{P(K+1)} = \underline{\mathbf{1+3+5+\ldots\ldots+(2K-1)}} + (2K+1)$

$= K^2 + (2K+1)$ *By assumption*

$= \mathbf{(K+1)^2}$ *By Perfect Square Equation*

So, p(K+1) is TRUE

Another Induction Example

Use mathematical Induction to prove that $N^3 - N$ is divisible by 3 whenever n is positive and n>=1

Basic step: P(1) is divisible by 3 since 1-1 = 0 and 3 divides 0

Inductive step:

A. Assume P(k) is true .

P(k): $K^3 - K$ is divisible by 3 $\Rightarrow$ $\underline{K^3\text{-}K= 3m}$, where m is an integer

B. Try to prove that p(k+1) is true as well

P(k+1): $(K+1)^3 - (K+1) = (K^3+3K^2+3K+1) - K - 1$

$= \underline{\mathbf{(K^3 - K)}} + 3(K^2 + K) = 3m + 3(K^2 + K) = 3(m + K^2 + K) = 3\ n$

So, p(k+1) is divisible by 3

Conclusion: $N^3 - N$ is divisible by 3 whenever n is positive integer

Module #9:
Basic Number Theory

3.1 The Integers and Division

- Of course you already know what the integers are, and what division is…
- **But:** There are some specific notations, terminology, and theorems associated with these concepts which you may not know.
- These form the basics of *number theory*.
  - Vital in many important algorithms today (hash functions, cryptography, digital signatures).

Divides, Factor, Multiple

- Let $a,b \in \mathbf{Z}$ with $a \neq 0$.
- $a|b \equiv$ "*a divides b*" $:\equiv$ "$\exists c \in \mathbf{Z}$: $b=ac$"
  "There is an integer c such that c times a equals b."
  - Example: $3|-12 \Leftrightarrow$ **True**, but $3|7 \Leftrightarrow$ **False**.
- Iff a divides b, then we say a is a *factor* or a *divisor* of b, and b is a *multiple* of a.
- "b is even" $:\equiv 2|b$. Is 0 even? Is -4?

Facts re: the Divides Relation

- $\forall\, a,b,c \in \mathbf{Z}$:
  1. $a|0$
  2. $(a|b \wedge a|c) \rightarrow a\,|\,(b+c)$
  3. $a|b \rightarrow a|bc$
  4. $(a|b \wedge b|c) \rightarrow a|c$
- **Proof** of (2): $a|b$ means there is an s such that $b=as$, and $a|c$ means that there is a t such that $c=at$, so $b+c = as+at = a(s+t)$, so $a|(b+c)$ also.■

More Detailed Version of Proof

- Show $\forall a,b,c \in \mathbf{Z}$: $(a|b \wedge a|c) \rightarrow a \mid (b + c)$.
- Let a, b, c be any integers such that $a|b$ and $a|c$, and show that $a \mid (b + c)$.
- By defn. of $|$, we know $\exists s$: $b=as$, and $\exists t$: $c=at$. Let s, t, be such integers.
- Then $b+c = as + at = a(s+t)$, so $\exists u$: $b+c=au$, namely $u=s+t$. Thus $a|(b+c)$.

Prime Numbers

- An integer $p>1$ is *prime* iff it is not the product of any two integers greater than 1:
  $$p>1 \wedge \neg\exists a,b\in\mathbf{N}\!: a>1,\ b>1,\ ab=p.$$
- The only positive factors of a prime 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.

Review

- $a|b \Leftrightarrow$ "*a divides b*" $\Leftrightarrow \exists c \in \mathbf{Z}: b = ac$
- "*p* is *prime*" $\Leftrightarrow$
  $p > 1 \wedge \neg\exists a \in \mathbf{N}: (1 < a < p \wedge a|p)$
- Terms *factor*, *divisor*, *multiple*, *composite*.

Fundamental Theorem of Arithmetic

Its "Prime Factorization"

- Every positive integer has a unique representation as the product of a non-decreasing series of zero or more primes.
  - 1 = (product of empty series) = 1
  - 2 = 2 (product of series with one element 2)
  - 4 = 2·2 (product of series 2,2)
  - 2000 = 2·2·2·2·5·5·5; 2001 = 3·23·29; 2002 = 2·7·11·13; 2003 = 2003

Theorem:

- Every positive integer greater than one can be **uniquely** written by one or more prime numbers.
- If n is composite integer, then n has prime divisor **less than or equal to**
- There are **infinitely** number of primes

"Prime Factorization Technique"

- To find the prime factor of an integer *n*:

1- find $\sqrt{n}$

2- list all primes <= $\sqrt{n}$

2, 3, 5, 7,…root of *n*

3- find all prime factors that divides *n*.

Ex: Show that 100 is composite?

Sol.

1) $\sqrt{100} = 10$

2) So the number may be divided by: **2, 3, 5, 7** only (all primes less than 10)

3) $2 \mid 100$ since $100/2 = 50$

$\therefore$ The number 100 is not prime, So it is composite.

Ex: Show that 101 is prime?

Sol.

1) $\sqrt{101} \approx 10$

2) So the number may be divided by: **2, 3, 5, 7** only (all primes less than 10)

3) $2 \nmid 101$ $\quad 3 \nmid 101$ $\quad 5 \nmid 101$ $\quad 7 \nmid 101$

101 is not divided by 2, 3, 4, 5, or 7

$\therefore$ The number 101 is prime

Ex: find the prime factors of 7007?

1) $\sqrt{7007} \approx 83$

2) So the number may be divided by: 2, 3, 5, 7, 11, 13, 17, 19 …< 83 (all primes less than 83)

3) $\dfrac{7007}{7} = 1001 \quad \dfrac{1001}{7} = 143 \quad \dfrac{143}{11} = 13 \quad \dfrac{13}{13} = 1$

$7007 = 7 \times 7 \times 11 \times 13 = 7^2 \times 11 \times 13$

Mersenne Primes

Any prime number that can be written as $2^P - 1$ is called Mersenne prime.

- **Ex:**

The numbers 2^2-1 =3, 2^3-1=7, 2^4-1=31 ………. 2^{11}-1 = 2047 all are **primes**

An Application of Primes

- When you visit a secure web site (`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!
  - **Note:** There is a tractable *quantum* algorithm for factoring; so if we can ever build big quantum computers, RSA will be insecure.

The Division "Algorithm"

- Really just a *theorem*, not an algorithm…
  - The name is used here for historical reasons.
- For any integer *dividend* a and *divisor* $d \neq 0$, there is a unique integer *quotient* q and *remainder* $r \in \mathbf{N} \ni a = dq + r$ and $0 \leq r < |d|$.
  (such that)
- $\forall a, d \in \mathbf{Z},\ d > 0\colon \exists!\, q, r \in \mathbf{Z}\colon 0 \leq r < |d|,\ a = dq + r.$
- We can find q and r by: $q = \lfloor a/d \rfloor$, $r = a - qd$.

The **mod** operator

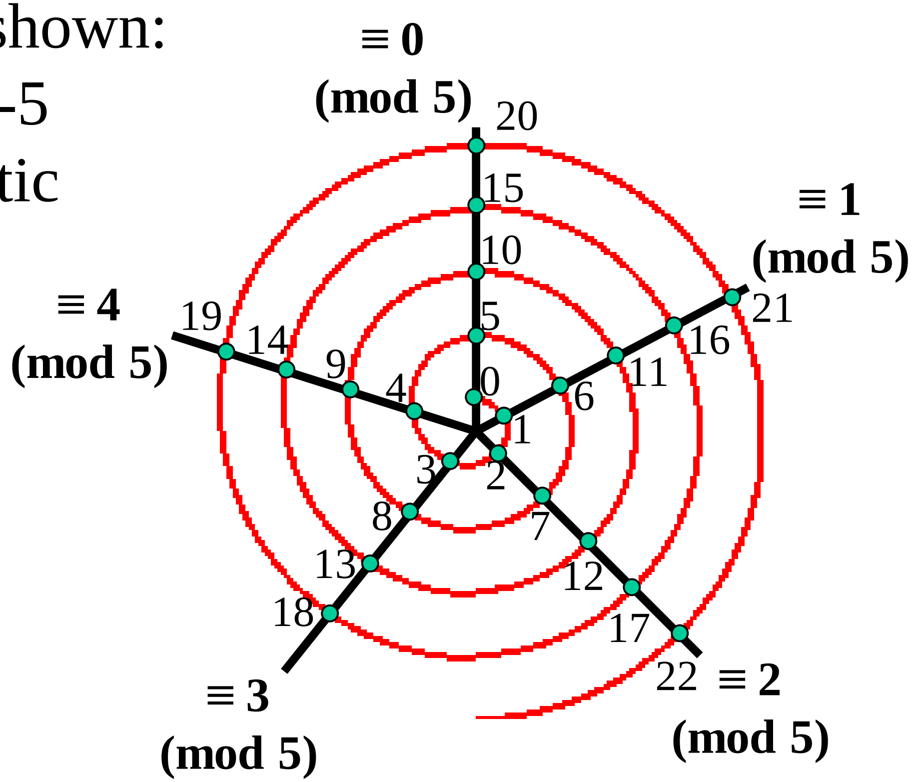
- An integer "division remainder" operator.
- Let $a,d\in\mathbf{Z}$ with $d>1$. Then a **mod** d denotes the remainder r from the division "algorithm" with dividend a and divisor d; *i.e.* the remainder when a is divided by d. (Using *e.g.* long division.)
- We can compute (a **mod** d) by: $a - d\cdot\lfloor a/d\rfloor$.
- In C programming language, "`%`" = mod.

Modular Congruence

- Let $\mathbf{Z}^+=\{n\in\mathbf{Z} \mid n>0\}$, the positive integers.
- Let $a,b\in\mathbf{Z}$, $m\in\mathbf{Z}^+$.
- Then *a is congruent to b modulo m*, written "$a\equiv b \pmod{m}$", iff $m \mid a-b$.
- Also equivalent to: $(a-b) \bmod m = 0$.
- (Note: this is a different use of "≡" than the meaning "is defined as" I've used before.)

Spiral Visualization of **mod**

Example shown:
modulo-5
arithmetic



Useful Congruence Theorems

- Let $a,b,c,d \in \mathbf{Z}$, $m \in \mathbf{Z}^+$. Then if $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, then:
  - $a+c \equiv b+d \pmod{m}$, and
  - $ac \equiv bd \pmod{m}$

Ex. :

a = 7 b = 2 c = 11

d = 1 m = 5

Since, $7 \equiv 2 \pmod 5$ where *mod* equals to 2 and

$11 \equiv 1 \pmod 5$ where *mod* equals to 1

Then, $7 + 11 \equiv 2 + 1 \pmod 5 \leftrightarrow 18 \equiv 3 \pmod 5$ where *mod* equals to 3

and, $7 \times 11 \equiv 2 \times 1 \pmod 5 \leftrightarrow 77 \equiv 2 \pmod 5$ where *mod* equals to 2

- **Thm:** Let **m** be a positive intger

  Integers **a**, and **b** are congruent modulo **m** iff **a = b + k m**, where **k** is an integer

Ex. :

- a = **17** b = **5** m = **6**

  Since, 17 mod 6 = 5

  and 5 mod 6 = 5

  Then, 17 ≡ 5 (mod 6) and 6 | (17-5) ↔ 6 | 12 where = **2**

  Also, **17 = 5 + 2 × 6**

Applications of Congruence

1. Hash Functions:

$h(k) = k \bmod m$ k: Key m: number of available memory locations

Notes:

Hash functions should be **onto**,

Since it **is not one-to-one**, this may cause **Collisions**.

Ex. : if $m = 50$, then $h(51) = h(101) = 1$

2. Pseudorandom Numbers:

To generate a sequence of random numbers $\{X_n\}$ with $0 \le X_n < m$

X_0: seed $\quad 0 \le X_0 < m$

$X_{n+1} = (a X_n + c) \bmod m$

Where, m: modulus

a: multiplier $\quad 2 \le a < m$

c: increment $\quad 0 \le c < m$

Ex. :

Given: $m = 9 \quad a = 7 \quad c = 4 \quad X_0=3$

Sequence:

$X_0 = 3$

$X_1 = (7 \times X_0+4) \bmod 9 = 7$

$X_2 = (7 \times X_1+4) \bmod 9 = 8$

$X_3 = 6$

$X_4 = 1$

.

.

$X_9 = 3$

3. Cryptology

Encryption: making a message secrete

Decryption: determining the original message

Ex. Caesar's Encryption

$f(x) = (x + \text{shift}) \bmod 26$

If shift = 3, then

The message: "***MEET YOU IN THE PARK***"

Becomes the encrypted message: "***PHHW BRX LG WKH SDUM***"

Since: A=0 becomes D=3, B=1 becomes E=4, …, X=23 becomes A=0, Y=24 becomes B=1, and finally Z=25 becomes C=2

3.2 Greatest Common Divisor

- The *greatest common divisor* gcd(a,b) of integers a,b (not both 0) is the largest (most positive) integer d that is a divisor both of a and of b.

  $d = \gcd(a,b) = \max(d: d|a \wedge d|b) \Leftrightarrow$
  $d|a \wedge d|b \wedge \forall e{\in}\mathbf{Z}, (e|a \wedge e|b) \rightarrow d \geq e$

- Example: gcd(24,36)=?
  Positive common divisors: 1,2,3,4,6,12…
  Greatest is 12.

Way to find GCD:

1. **find all positive common divisors of both a and b, then take the largest divisor**

Ex: find gcd (24, 36)?

Divisors of 24: 1, 2, 3, 4, 6, 8, 12, 24

Divisors of 36: 1, 2, 3, 4, 6, 8, 12, 18, 24

Common divisors: 1, 2, 3, 4, 6, 8, 12

MAXIMUM = 12

∴ gcd (24, 36) = 12

2. use prime factorization:

- If the prime factorizations are written as
  $a = p_1^{a_1} p_2^{a_2} \ldots p_n^{a_n}$ and $b = p_1^{b_1} p_2^{b_2} \ldots p_n^{b_n}$
  then the GCD is given by:

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

- Example:
  - a=84=2·2·3·7 $= 2^2 \cdot 3^1 \cdot 7^1$
  - b=96=2·2·2·2·2·3 $= 2^5 \cdot 3^1 \cdot 7^0$
  - gcd(84,96) $= 2^2 \cdot 3^1 \cdot 7^0 = 2 \cdot 2 \cdot 3 = 12$.

Ex: find gcd (24, 36)?

$24 = 2^3 \times 3^1$

$36 = 2^2 \times 3^2$

$\therefore$ gcd (24, 36) $= 2^2 \times 3^1 = 12$

Ex: find gcd (120, 500)?

$120 = 2^3 \times 3 \times 5$

$36 = 2^2 \times 5^3 = 2^2 \times 3^0 \times 5^3$

$\therefore$ gcd (120, 500) $= 2^2 \times 5 = 20$

Relatively Prime

- Integers *a* and *b* are called *relatively prime* or *coprime* iff their gcd = 1.
  - Example: Neither 21 and 10 are prime, but they are *relatively prime*. $21=3\cdot7$ and $10=2\cdot5$, so they have no common factors > 1, so their gcd = 1.
- A *set* of integers $\{a_1,a_2,\ldots\}$ is *(pairwise) relatively prime* if all pairs a_i, a_j, $i\neq j$, are relatively prime.

Least Common Multiple

- 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.* lcm(6,10)=30

  $m = \mathrm{lcm}(a,b) = \min(m{:}\ a|m \wedge b|m) \Leftrightarrow$
  $a|m \wedge b|m \wedge \forall n\in\mathbf{Z}{:}\ (a|n \wedge b|n) \rightarrow (m \leq n)$

- If the prime factorizations are written as

  $a = p_1^{a_1} p_2^{a_2} \ldots p_n^{a_n}$ and $b = p_1^{b_1} p_2^{b_2} \ldots p_n^{b_n}$;

  then the LCM is given by

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

Ex: find Lcm (24, 36)?

$24 = 2^3 \times 3^1$

$36 = 2^2 \times 3^2$

$\therefore$ Lcm (24, 36) $= 2^3 \times 3^2 = 72$

Ex: find Lcm (120, 500)?

$120 = 2^3 \times 3 \times 5$

$36 = 2^2 \times 5^3$

$\therefore$ Lcm (120, 500) $= 2^3 \times 3 \times 5^3 = 3000$

3.3 Matrices

- A *matrix* is a rectangular array of objects (usually numbers).
- An $m \times n$ ("m by n") matrix has exactly m horizontal rows, and n vertical columns.
- Plural of matrix = *matrices* (say MAY-trih-sees)
- An $n \times n$ matrix is called a *square* matrix, whose *order* is n.

$$\begin{bmatrix} 2 & 3 \\ 5 & -1 \\ 7 & 0 \end{bmatrix}$$

a 3×2 matrix

Applications of Matrices

Tons of applications, including:

- Solving systems of linear equations
- Computer Graphics, Image Processing
- Models within many areas of Computational Science & Engineering
- Quantum Mechanics, Quantum Computing
- Many, many more…

Matrix Equality

- Two matrices **A** and **B** are equal iff they have the same number of rows, the same number of columns, and all corresponding elements are equal.

$$\begin{bmatrix} 3 & 2 \\ -1 & 6 \end{bmatrix} \neq \begin{bmatrix} 3 & 2 & 0 \\ -1 & 6 & 0 \end{bmatrix}$$

Row and Column Order

- The rows in a matrix are usually indexed 1 to *m* from top to bottom. The columns are usually indexed 1 to *n* from left to right. Elements are indexed by row, then column.

$$\mathbf{A} = [a_{i,j}] = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix}$$

Matrix Sums

- The *sum* **A**+**B** of two matrices **A**, **B** (which **must** have the same number of rows, and the same number of columns) is the matrix (also with the same shape) given by adding corresponding elements.
- **A**+**B** = $[a_{i,j}+b_{i,j}]$

$$\begin{bmatrix} 2 & 6 \\ 0 & -8 \end{bmatrix} + \begin{bmatrix} 9 & 3 \\ -11 & 3 \end{bmatrix} = \begin{bmatrix} 11 & 9 \\ -11 & -5 \end{bmatrix}$$

Matrix Products

- For an $m\times k$ matrix **A** and a $k\times n$ matrix **B**, the *product* **AB** is the $m\times n$ matrix:

$$\mathbf{AB} = \mathbf{C} = [c_{i,j}] \equiv \left[\sum_{\ell=1}^{k} a_{i,\ell} b_{\ell,j} \right]$$

- *I.e.*, element (i,j) of **AB** is given by the vector *dot product* of the <u>ith row of **A**</u> and the <u>jth column of **B**</u> (considered as vectors).
- Note: Matrix multiplication is **not** commutative!

Matrix Product Example

- An example matrix multiplication to practice in class:

$$\begin{bmatrix} 0 & 1 & -1 \\ 2 & 0 & 3 \end{bmatrix} \cdot \begin{bmatrix} 0 & -1 & 1 & 0 \\ 2 & 0 & -2 & 0 \\ 1 & 0 & 3 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -5 & -1 \\ 3 & -2 & 11 & 3 \end{bmatrix}$$

Identity Matrices

- The *identity matrix of order n*, $\mathbf{I}_n$, is the order-*n* matrix with 1's along the upper-left to lower-right diagonal and 0's everywhere else.

$$\boldsymbol{I}_n = \left[\begin{cases} 1 \text{ if } i = j \\ 0 \text{ if } i \neq j \end{cases}\right] = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

Review

Matrix sums and products:

$$\mathbf{A}+\mathbf{B} = [a_{i,j}+b_{i,j}]$$

$$\mathbf{AB} = \mathbf{C} = [c_{i,j}] \equiv \left[\sum_{\ell=1}^{k} a_{i,\ell} b_{\ell,j} \right]$$

Identity matrix of order *n*:

$\mathbf{I}_n = [\delta_{ij}]$, where $\delta_{ij}=1$ if $i=j$ and $\delta_{ij}=0$ if $i\neq j$.

Matrix Inverses

- For some (but not all) square matrices $\mathbf{A}$, there exists a unique multiplicative *inverse* $\mathbf{A}^{-1}$ of $\mathbf{A}$, a matrix such that $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}_n$.
- If the inverse exists, it is unique, and $\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1}$.
- We won’t go into the algorithms for matrix inversion...

Matrix Multiplication Algorithm

procedure *matmul*(matrices **A**: $m \times k$, **B**: $k \times n$)

for i := 1 **to** m $\Theta(m)\cdot$

  for j := 1 **to** n **begin** $\Theta(n)\cdot($

    c_{ij} := 0 $\Theta(1)+$

    for q := 1 **to** k $\Theta(k)\cdot$

      $c_{ij} := c_{ij} + a_{iq}b_{qj}$ $\Theta(1))$

  end {**C**=[c_{ij}] is the product of **A** and **B**}

What's the Θ of its time complexity?

Answer: $\Theta(mnk)$

Powers of Matrices

If **A** is an $n \times n$ square matrix and $p \geq 0$, then:

- $\mathbf{A}^p \equiv \underbrace{\mathbf{AAA}\cdots\mathbf{A}}_{p\text{ times}} \quad (\mathbf{A}^0 \equiv \mathbf{I}_n)$
- Example:

$$\begin{bmatrix} 2 & 1 \\ -1 & 0 \end{bmatrix}^3 = \begin{bmatrix} 2 & 1 \\ -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 2 & 1 \\ -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 2 & 1 \\ -1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 1 \\ -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 3 & 2 \\ -2 & -1 \end{bmatrix}$$

$$= \begin{bmatrix} 4 & 3 \\ -3 & -2 \end{bmatrix}$$

Matrix Transposition

- If $\mathbf{A}=[a_{ij}]$ is an $m \times n$ matrix, the *transpose* of $\mathbf{A}$ (often written $\mathbf{A}^t$ or $\mathbf{A}^T$) is the $n \times m$ matrix given by $\mathbf{A}^t = \mathbf{B} = [b_{ij}] = [a_{ji}]$ $(1 \le i \le n, 1 \le j \le m)$

$$\begin{bmatrix} 2 & 1 & 3 \\ 0 & -1 & -2 \end{bmatrix}^t = \begin{bmatrix} 2 & 0 \\ 1 & -1 \\ 3 & -2 \end{bmatrix}$$

Flip across diagonal

Symmetric Matrices

- A square matrix **A** is *symmetric* iff **A**=**A**^{t}. *I.e.*, $\forall i,j \le n$: $a_{ij} = a_{ji}$.
- Which is symmetric?

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} -2 & 1 & 3 \\ 1 & 0 & -1 \\ 3 & -1 & 2 \end{bmatrix} \begin{bmatrix} 3 & 0 & 1 \\ 0 & 2 & -1 \\ 1 & 1 & -2 \end{bmatrix}$$

Zero-One Matrices

- Useful for representing other structures.
  - *E.g.*, relations, directed graphs (later in course)
- All elements of a *zero-one* matrix are 0 or 1
  - Representing **False** & **True** respectively.
- The *join* of **A**, **B** (both $m \times n$ zero-one matrices):
  - $\mathbf{A} \wedge \mathbf{B} :\equiv [a_{ij} \wedge b_{ij}] = [a_{ij}\, b_{ij}]$
- The *meet* of **A**, **B**:
  - $\mathbf{A} \vee \mathbf{B} :\equiv [a_{ij} \vee b_{ij}]$

Join ($\vee$)

A = B=

We find the join between $A \vee B =$

Meet $\wedge$

We find the join between $A \wedge B =$

Boolean Products

- Let $\mathbf{A}=[a_{ij}]$ be an $m\times k$ zero-one matrix, & let $\mathbf{B}=[b_{ij}]$ be a $k\times n$ zero-one matrix,
- The *boolean product* of **A** and **B** is like normal matrix $\times$, but using $\vee$ instead + in the row-column "vector dot product."

$$\mathbf{A}\odot\mathbf{B} = \mathbf{C} = [c_{ij}] = \left[\bigvee_{\ell=1}^{k} a_{i\ell} \wedge b_{\ell j}\right]$$

Boolean Powers

- For a square zero-one matrix **A**, and any $k \geq 0$, the *kth Boolean power of **A*** is simply the Boolean product of k copies of **A**.
- $\mathbf{A}^{[k]} \equiv \underbrace{\mathbf{A} \odot \mathbf{A} \odot \ldots \odot \mathbf{A}}_{k \text{ times}}$

Module #12:
Sequences

§3.2: Sequences, Strings, & Summations

- A *sequence* or *series* is just like an ordered *n*-tuple, except:
  - Each element in the series has an associated *index* number.
  - A sequence or series may be infinite.
- A *string* is a sequence of *symbols* from some finite *alphabet*.
- A *summation* is a compact notation for the sum of all terms in a (possibly infinite) series.

Sequences

- A *sequence* or *series* $\{a_n\}$ is identified with a *generating function* $f:S\rightarrow A$ for some subset $S\subseteq\mathbf{N}$ and for some set A.
  - Often we have $S=\mathbf{N}$ or $S=\mathbf{N}+\{0\}$.
  - Sequences may also be generalized to *indexed sets*, in which the set S does *not* have to be a subset of **N**.
    - For general indexed sets, S may not even be a set of numbers at all.
- If f is a generating function for a series $\{a_n\}$, then for $n\in S$, the symbol a_n denotes $f(n)$, also called *term n* of the sequence.
  - The *index* of a_n is n. (Or, often i is used.)
- A series is sometimes denoted by listing its first and/or last few elements, and using ellipsis (…) notation.
  - E.g., "$\{a_n\} = 0, 1, 4, 9, 16, 25, \ldots$" is taken to mean $\forall n\in\mathbf{N}$, $a_n = n^2$.

Sequence Examples

- Some authors write "the sequence $a_1, a_2, \ldots$" instead of $\{a_n\}$, to ensure that the set of indices is clear.
  - Be careful: Our book often leaves the indices ambiguous.
- An example of an infinite series:
  - Consider the series $\{a_n\} = a_1, a_2, \ldots$, where $(\forall n \geq 1)$ $a_n = f(n) = 1/n$.
  - Then, we have $\{a_n\} = 1, 1/2, 1/3, \ldots$

Example with Repetitions

- Like tuples, but unlike sets, a sequence may contain repeated instances of an element.
- Consider the sequence $\{b_n\} = b_0, b_1, \ldots$ (note that 0 is an index) where $b_n = (-1)^n$.
  - Thus, $\{b_n\} = 1, -1, 1, -1, \ldots$
    - Note repetitions!
  - This $\{b_n\}$ denotes an infinite sequence of 1's and –1's, *not* the 2-element set $\{1, -1\}$.

Sequences are two types:

- **Geometric progression**: it is a sequence of form

  $a, \; ar, ar^2, \; ar^3 \; \ldots\ldots\ldots, ar^n$

  where the *initial term* is **a** and the *common ratio* **r** are real numbers.

- **Arithmetic progression**: it is a sequence of form

  $a, a+d, a+2d, \ldots\ldots\ldots, \; a + nd$

  where the *initial term* **a** and the *common difference* **d** are real numbers

Examples of Geometric

- $\{b_n\}$ with $b_n = (-1)^n$ $n>=1$
- A_n = -1, 1, -1, 1,……..
  - initial term = -1, common ratio= -1,
  - $a_n=(-1)^n$, n=1,2,3,…
- C_n = 10,50,250,1250,……….
  - initial term = 10, common ratio= 5
  - $a_n=10\times (5)^n$, n=0,1,2,3,…
  - $a_n=5 \text{ x } a_{n-1}$, n=0,1,2,3,…

Examples of Arithmetic

- {Sn} with $S_n = -1 + 4n, n \geq 0$

  is arithmetic sequence where

  S_n = -1, 3, 7, 11,… OR $S_n = a_{n-1}+4, S_0=-1, n \geq 0$

  - initial term = -1, common Difference= 4
- The sequence : 5, 11, 17, 23, 29 …

  $a_n = 6n - 1 \quad , n \geq 1$

  is arithmetic progression with a= 5, and d= 6

Recognizing Sequences

- Sometimes, you're given the first few terms of a sequence,
  - and you are asked to find the sequence's generating function,
  - or a procedure to enumerate the sequence.
- Examples: What's the next number?
  - 1,2,3,4,… 5 (the 5th smallest number >0)
  - 1,3,5,7,9,… 11 (the 6th smallest odd number >0)
  - 2,3,5,7,11,... 13 (the 6th smallest prime number)

The Trouble with Sequence Recognition

- As you know, these problems are popular on IQ tests, but…
- The problem of finding "the" generating function given just an initial subsequence is *not a mathematically well defined problem.*
  - This is because there are *infinitely* many computable functions that will generate *any* given initial subsequence.
- We implicitly are supposed to find the *simplest* such function (because this one is assumed to be most likely), but,
  - how are we to objectively define the *simplicity* of a function?
- We might define simplicity as the reciprocal of complexity, but…
  - There are *many* different plausible, competing definitions of complexity, and this is an active research area.
- So, these questions really have *no* objective right answer!
  - Still, we will ask you to answer them anyway… (Because others will too.)

Strings, more formally

- Let Σ be a finite set of *symbols*, *i.e.* an *alphabet.*
  - A *string s* over alphabet Σ is any sequence $\{s_i\}$ of symbols, $s_i \in \Sigma$, indexed by **N** or **N**–{0}.
- If *a*, *b*, *c*, … are symbols, the string *s* = *a*, *b*, *c*, … can also be written *abc* …(*i.e.*, without commas).
- If *s* is a finite string and *t* is any string, then the *concatenation of s with t*, written just *st,*
  - is simply the string consisting of the symbols in *s*, in sequence, followed by the symbols in *t*, in sequence.

More Common String Notations

- The length $|s|$ of a finite string s is its number of *positions* (*i.e.*, its number of index values i).
- If s is a finite string and $n \in \mathbf{N}$,
  - Then s^n denotes the concatenation of n copies of s.
- ε denotes the empty string, the string of length 0.

Module #13:
Summations

Summation Notation

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

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

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

Generalized Summations

- For an infinite series, we may write:

$$\sum_{i=j}^{\infty} a_i :\equiv a_j + a_{j+1} + \ldots$$

- To sum a function over all members of a set $X=\{x_1, x_2, \ldots\}$: $\sum_{x \in X} f(x) :\equiv f(x_1) + f(x_2) + \ldots$
- Or, if $X=\{x|P(x)\}$, we may just write:

$$\sum_{P(x)} f(x) :\equiv f(x_1) + f(x_2) + \ldots$$

Simple Summation Example

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

$$= (4+1)+(9+1)+(16+1)$$

$$= 5+10+17$$

$$= 32$$

More Summation Examples

- An infinite series with a finite sum:

$$\sum_{i=0}^{\infty} 2^{-i} = 2^0 + 2^{-1} + \ldots = 1 + \tfrac{1}{2} + \tfrac{1}{4} + \ldots = 2$$

- 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:

$$\sum_x cf(x) = c\sum_x f(x)$$

(Distributive law.)

$$\sum_x f(x) + g(x) = \left(\sum_x f(x)\right) + \sum_x g(x)$$

(Application of commut-ativity.)

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

(Index shifting.)

More Summation Manipulations

- Other identities that are sometimes useful:

(Series splitting.)

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

$$\sum_{i=0}^{2k} f(i) = \sum_{i=0}^{k} f(2i) + f(2i+1)$$ (Grouping.)

Nested Summations

- These have the meaning you'd expect.

$$\sum_{i=1}^{4}\sum_{j=1}^{3} ij = \sum_{i=1}^{4}\left(\sum_{j=1}^{3} ij\right) = \sum_{i=1}^{4} i\left(\sum_{j=1}^{3} j\right) = \sum_{i=1}^{4} i(1+2+3)$$

$$= \sum_{i=1}^{4} 6i = 6\sum_{i=1}^{4} i = 6(1+2+3+4)$$

$$= 6 \cdot 10 = 60$$

- Note issues of free vs. bound variables, just like in quantified expressions, integrals, etc.

Some Shortcut Expressions(1)

$$\sum_{k=0}^{n} ar^k = a(r^{n+1}-1)/(r-1), r \neq 1$$ Geometric series.

$$\sum_{k=1}^{n} k = n(n+1)/2$$ Euler's trick.

$$\sum_{k=1}^{n} k^2 = n(n+1)(2n+1)/6$$ Quadratic series.

$$\sum_{k=1}^{n} k^3 = n^2(n+1)^2/4$$ Cubic series.

Some Shortcut Expressions(2)

$$\sum_{k=0}^{\infty} x^k = 1/(1-x), \; |x| < 1$$

$$\sum_{k=1}^{\infty} kx^{k-1} = 1/(1-x)^2, \; |x| < 1$$

Using the Shortcuts

- Example: Evaluate $\sum_{k=50}^{100} k^2$.
  - Use series splitting.
  - Solve for desired summation.
  - Apply quadratic series rule.
  - Evaluate.

$$\sum_{k=1}^{100} k^2 = \left(\sum_{k=1}^{49} k^2\right) + \sum_{k=50}^{100} k^2$$

$$\sum_{k=50}^{100} k^2 = \left(\sum_{k=1}^{100} k^2\right) - \sum_{k=1}^{49} k^2$$

$$= \frac{100 \cdot 101 \cdot 201}{6} - \frac{49 \cdot 50 \cdot 99}{6}$$

$$= 338{,}350 - 40{,}425$$

$$= 297{,}925.$$

Example

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

$$\sum_{k=1}^{100} k^2 = \left(\sum_{k=1}^{49} k^2\right) + \sum_{k=50}^{100} k^2$$

$$\sum_{k=50}^{100} k^2 = \left(\sum_{k=1}^{100} k^2\right) - \sum_{k=1}^{49} k^2$$

$$= \frac{100 \cdot 101 \cdot 201}{6} - \frac{49 \cdot 50 \cdot 99}{6}$$

$$= 338{,}350 - 40{,}425$$

$$= 297{,}925$$

Summations: Conclusion

- You need to know:
  - How to read, write & evaluate summation expressions like:

$$\sum_{i=j}^{k} a_i \qquad \sum_{i=j}^{\infty} a_i \qquad \sum_{x \in X} f(x) \qquad \sum_{P(x)} f(x)$$

  - Summation manipulation laws we covered.
  - Shortcut closed-form formulas, & how to use them.

Module #4:
Functions

Section 2.3… Functions

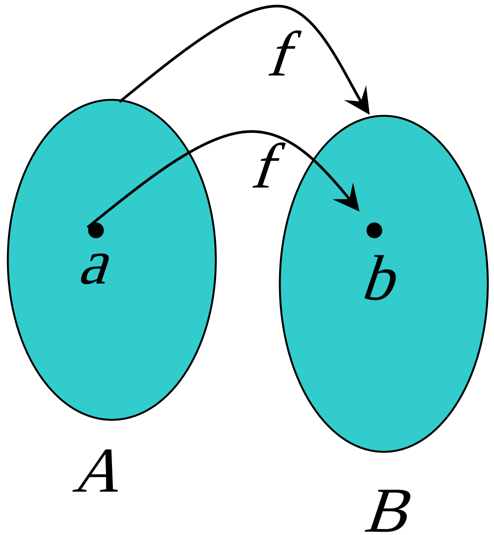
- From calculus, you are familiar with the concept of a real-valued function f, which assigns to each number $x \in \mathbf{R}$ a particular value $y = f(x)$, where $y \in \mathbf{R}$.
- But, the notion of a function can also be naturally generalized to the concept of assigning elements of *any* set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

Function: Formal Definition

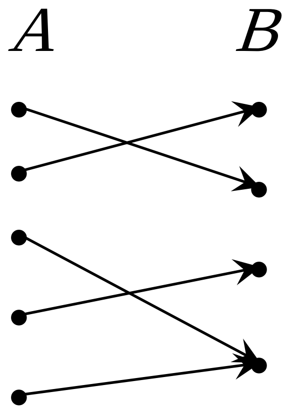
- For any sets A, B, we say that a *function f from (or "mapping") A to B* ($f{:}A\rightarrow B$) is a particular assignment of exactly **<u>one</u>** element $f(x)\in B$ to **<u>each</u>** element $x\in A$.
- Some further generalizations of this idea:
  - A *partial* (non-*total*) function f assigns *zero or one* elements of B to each element $x\in A$.
  - Functions of n arguments; relations (ch. 6).

Graphical Representations

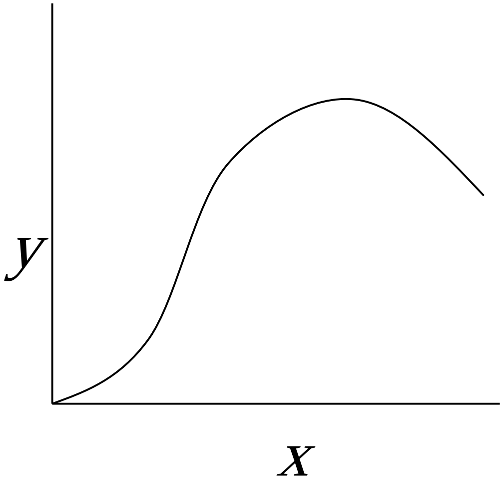
- Functions can be represented graphically in several ways:



Like Venn diagrams



2-part Graph



Plot

Some Function Terminology

- If it is written that $f{:}A\rightarrow B$, and $f(a)=b$ (where $a\in A$ & $b\in B$), then we say:
  - A is the *domain* of f.
  - B is the *codomain* of f.
  - b is the *image* of a under f.
  - a is a *pre-image* of b under f.
    - In general, 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 to you that: "f is a function mapping students in this class to the set of grades {A,B,C,D,E}."
- At this point, you know f's codomain is: {A,B,C,D,E}, 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,E}!.

Constructing Function Operators

- If • ("dot") is any operator over B, then we can extend • to also denote an operator over functions $f{:}A\rightarrow B$.
- *E.g.*: Given any binary operator $\bullet{:}B\times B\rightarrow B$, and functions $f,g{:}A\rightarrow B$, we define $(f\bullet g){:}A\rightarrow B$ to be the function defined by: $\forall a\in A,\ (f\bullet g)(a) = f(a)\bullet g(a)$.

Function Operator Example

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

Function Composition Operator

Note match here.

- 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 out of f and g by applying f to the result of applying g.
  - We say $(f\circ g)$:$A\rightarrow C$, where $(f\circ g)(a) :\equiv f(g(a))$.
  - Note $g(a)\in B$, so $f(g(a))$ is defined and $\in C$.
  - Note that $\circ$ (like Cartesian $\times$, but unlike $+,\wedge,\cup$) is non-commuting. (Generally, $f\circ g \neq g\circ f$.)

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.

$$f(S) :\equiv \{f(s) \mid s\in S\}$$
$$:\equiv \{b \mid \exists\, s\in S\colon f(s)=b\}.$$

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

One-to-One Functions

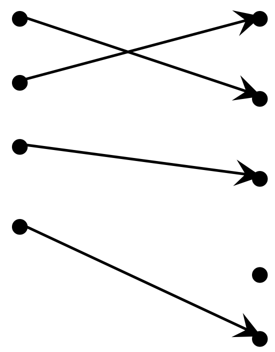
- A function is *one-to-one (1-1)*, or *injective*, or *an injection*, iff every element of its range has *only* 1 pre-image.
  - Formally: given $f:A\rightarrow B$,
    "x is injective" $:\equiv (\neg\exists x,y\colon x\neq y \wedge f(x)=f(y))$.
- Only one element of the domain is mapped to any given one element of the range.
  - Domain & range have same cardinality. What about codomain?

May Be Larger

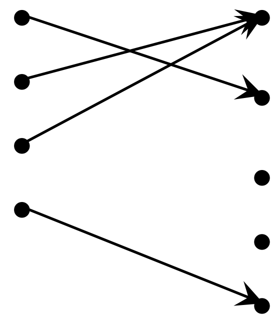
- Each element of the domain is injected into a different element of the range.
  - Compare "each dose of vaccine is injected into a different patient."

One-to-One Illustration

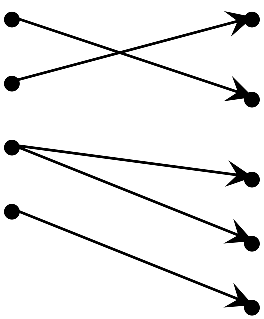
- (2-part) graph representations of functions that are (or not) one-to-one:



One-to-one



Not one-to-one



Not even a function!

Examples

- f: Z → Z $\qquad f(x)=x^2$

  $f(x)=f(y) \Rightarrow x^2 = y^2 \Rightarrow x=+y$ or $x=-y$

  $f(-2)=f(2)=4 \Rightarrow -2\neq 2 \Rightarrow$ it is not 1-to-1

- f: Z → Z $\qquad f(x)=x+5$

  $f(x)=f(y) \Rightarrow x+5=y+5 \Rightarrow x=y$

  $\Rightarrow$ it is 1-to-1

Sufficient Conditions for 1-1ness

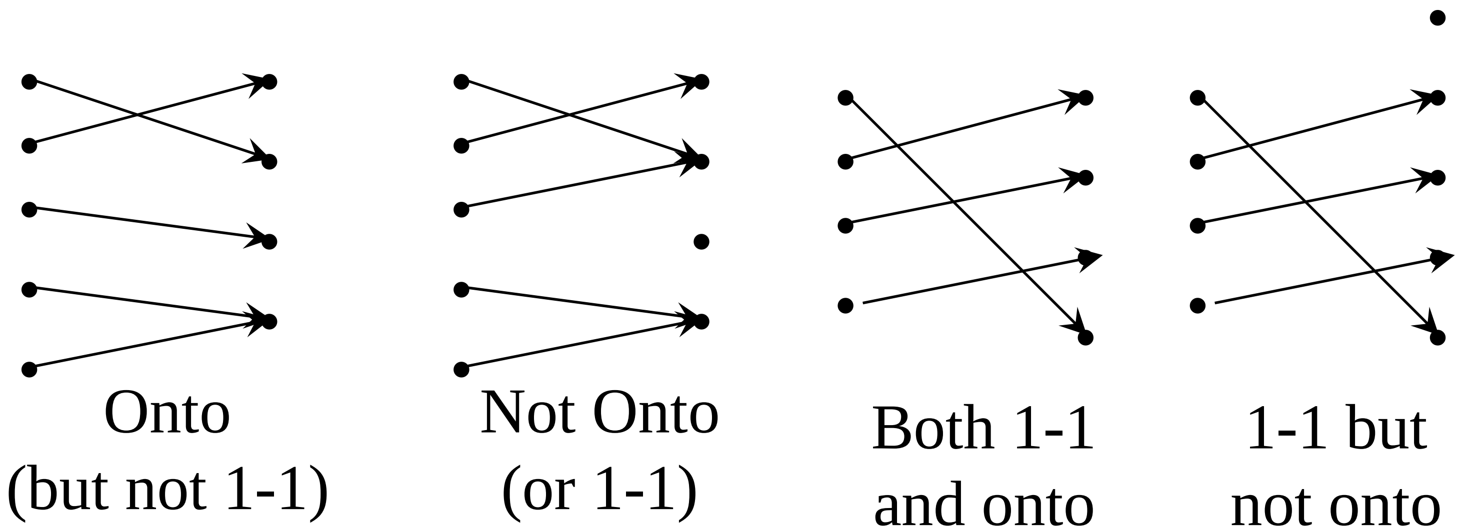
- For functions f over numbers, we say:
  - f is *strictly* (or *monotonically*) *increasing* iff $x>y \rightarrow f(x)>f(y)$ for all x,y in domain;
  - f is *strictly* (or *monotonically*) *decreasing* iff $x>y \rightarrow f(x)<f(y)$ for all x,y in domain;
- If f is either strictly increasing or strictly decreasing, then f is one-to-one. *E.g.* x^3
  - *Converse is not necessarily true. E.g.* $1/x$

Onto (Surjective) Functions

- A function $f:A\rightarrow B$ is *onto* or *surjective* or *a surjection* iff its range is equal to its codomain ($\forall b\in B, \exists a\in A: f(a)=b$).
- 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 **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:



Bijections

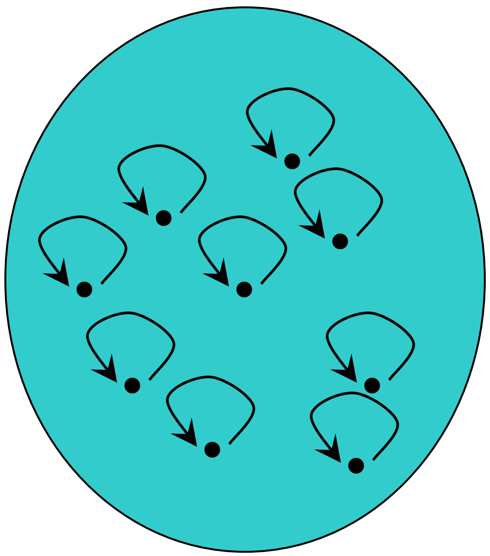
- A function f is said to be *a one-to-one correspondence*, or *a bijection*, or *reversible*, or *invertible*, iff it is both one-to-one and onto.
- For bijections $f{:}A\rightarrow B$, there exists an *inverse of f*, written $f^{-1}{:}B\rightarrow A$, which is the unique function such that $f^{-1}\circ f = I_A$
  - (where I_A is the identity function on A)

The Identity Function

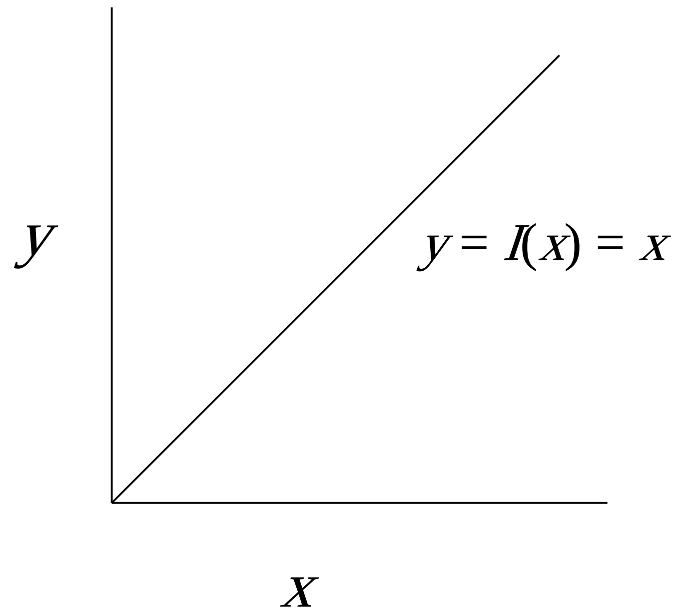
- For any domain A, the *identity function* $I{:}A \rightarrow A$ (variously written, I_A, $\mathbf{1}$, $\mathbf{1}_A$) is the unique function such that $\forall a \in A{:}\ I(a)=a$.
- Some identity functions you've seen:
  - +ing 0, ·ing by 1, $\wedge$ing with **T**, $\vee$ing with **F**, $\cup$ing with $\varnothing$, $\cap$ing with U.
- Note that the identity function is always both one-to-one and onto (bijective).

Identity Function Illustrations

- The identity function:



Domain and range



Graphs of Functions

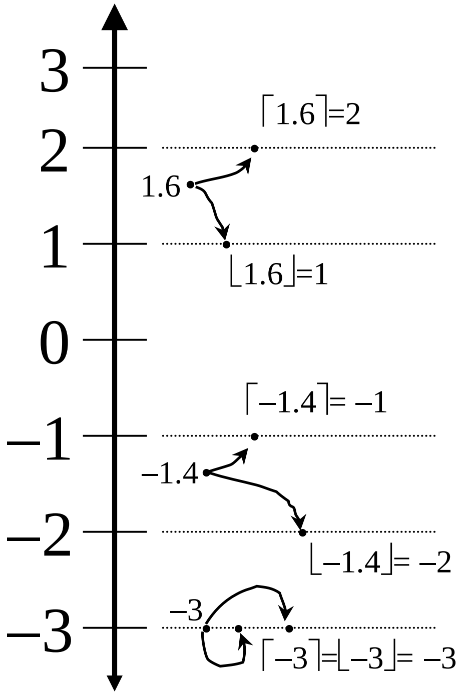
- We can represent a function $f:A\rightarrow B$ as a set of ordered pairs $\{(a,f(a)) \mid a\in A\}$. ← The function's *graph*.
- Note that $\forall a$, there is only 1 pair (a,b).
  - Later (ch.6): *relations* loosen this restriction.
- For functions over numbers, we can represent an ordered pair (x,y) as a point on a plane.
  - A function is then drawn as a curve (set of points), with only one y for each x.

A Couple of Key Functions

- In discrete math, we will frequently use the following two functions over real numbers:
  - The *floor* function $\lfloor\cdot\rfloor$:**R**→**Z**, where $\lfloor x\rfloor$ ("floor of *x*") means the largest (most positive) integer $\leq x$. *I.e.*, $\lfloor x\rfloor :\equiv \max(\{i\in\mathbf{Z} \mid i\leq x\})$.
  - The *ceiling* function $\lceil\cdot\rceil$:**R**→**Z**, where $\lceil x\rceil$ ("ceiling of *x*") means the smallest (most negative) integer $\geq x$. $\lfloor x\rfloor :\equiv \min(\{i\in\mathbf{Z} \mid i\geq x\})$

Visualizing Floor & Ceiling

- Real numbers “fall to their floor” or “rise to their ceiling.”
- Note that if $x \notin \mathbf{Z}$, $\lfloor -x \rfloor \neq -\lfloor x \rfloor$ & $\lceil -x \rceil \neq -\lceil x \rceil$
- Note that if $x \in \mathbf{Z}$, $\lfloor x \rfloor = \lceil x \rceil = x$.

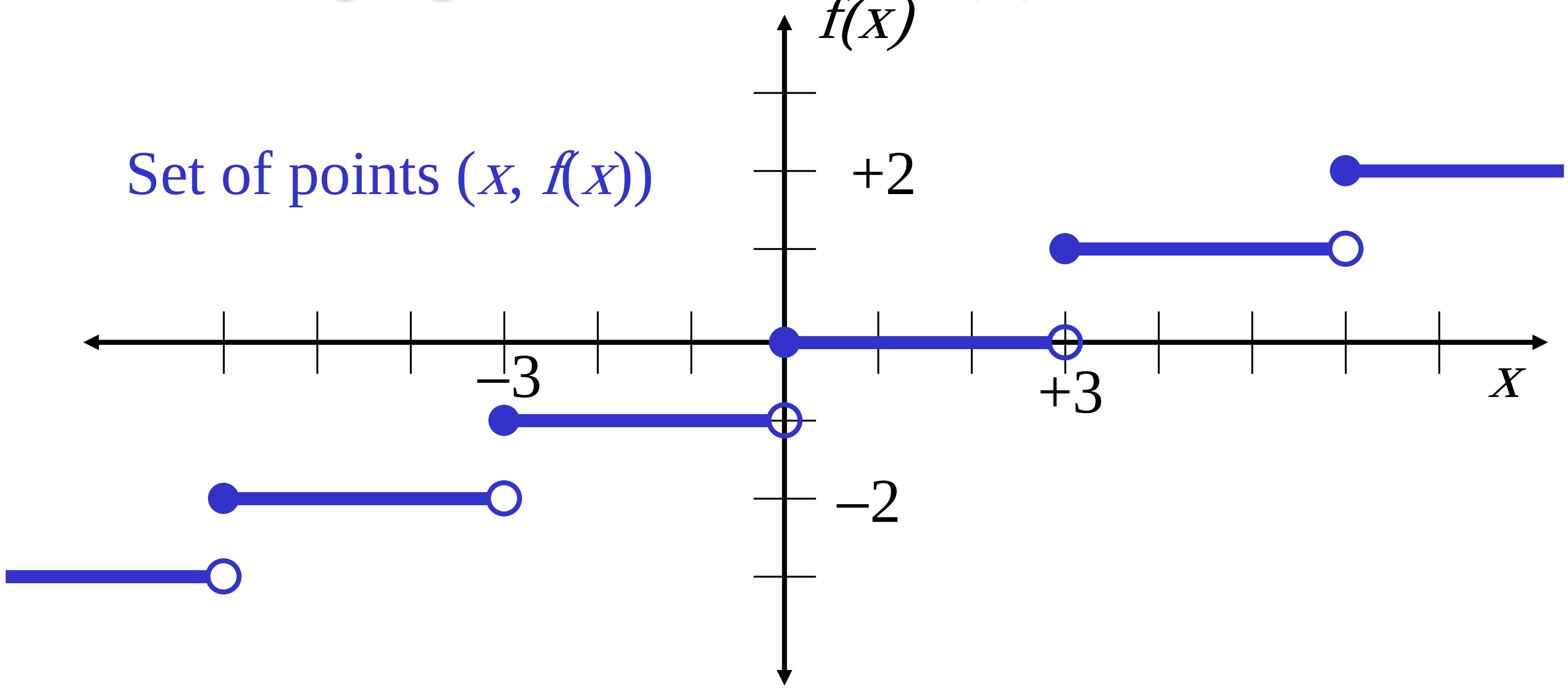


Plots with floor/ceiling

- Note that for $f(x)=\lfloor x \rfloor$, the graph of f includes the point $(a, 0)$ for all values of a such that $a \geq 0$ and $a < 1$, but not for the value $a=1$.
- We say that the set of points $(a,0)$ that is in f does not include its *limit* or *boundary* point $(a,1)$.
  - Sets that do not include all of their limit points are generally called *open sets*.
- In a plot, we draw a limit point of a curve using an open dot (circle) if the limit point is not on the curve, and with a closed (solid) dot if it is on the curve.

Plots with floor/ceiling: Example

- Plot of graph of function $f(x) = \lfloor x/3 \rfloor$:



Review of §2.3 (Functions)

- Function variables f, g, h, …
- Notations: $f:A\rightarrow B$, $f(a)$, $f(A)$.
- Terms: image, preimage, domain, codomain, range, one-to-one, onto, strictly (in/de)creasing, bijective, inverse, composition.
- Function unary operator f^{-1}, binary operators $+$, $-$, *etc.*, and $\circ$.
- The $\mathbf{R}\rightarrow\mathbf{Z}$ functions $\lfloor x \rfloor$ and $\lceil x \rceil$.

Relations

Binary Relations

- Let A, B be any two sets.
- A *binary relation R* from A to B, written (with signature) $R{:}A\leftrightarrow B$, is a subset of $A\times B$.
  - *E.g.*, let $< : \mathbf{N}\leftrightarrow\mathbf{N} :\equiv \{(n,m) \mid n < m\}$
- The notation $a\ R\ b$ or aRb means $(a,b)\in R$.
  - *E.g.*, $a < b$ means $(a,b)\in <$
- If aRb we may say "a is related to b (by relation R)", or "a relates to b (under relation R)".
- A binary relation R corresponds to a predicate function $P_R{:}A\times B\rightarrow\{\mathbf{T},\mathbf{F}\}$ defined over the 2 sets A,B; *e.g.*, "eats" $:\equiv \{(a,b) \mid$ organism a eats food $b\}$

Complementary Relations

- Let R:$A\leftrightarrow B$ be any binary relation.
- Then, $\cancel{R}$:$A\leftrightarrow B$, the *complement* of R, is the binary relation defined by
  $$\cancel{R} :\equiv \{(a,b) \mid (a,b)\notin R\} = (A\times B) - R$$
  Note this is just $\overline{R}$ if the universe of discourse is $U = A\times B$; thus the name *complement.*
- Note the complement of $\cancel{R}$ is R.

Example: $\not< \; = \{(a,b) \mid (a,b)\notin <\} = \{(a,b) \mid \neg a<b\} = \geq$

Inverse Relations

- Any binary relation $R{:}A\leftrightarrow B$ has an *inverse* relation $R^{-1}{:}B\leftrightarrow A$, defined by
  $$R^{-1} :\equiv \{(b,a) \mid (a,b)\in R\}.$$
  E.g., $<^{-1} = \{(b,a) \mid a<b\} = \{(b,a) \mid b>a\} = >$.
- *E.g.*, if R:People→Foods is defined by $aRb \Leftrightarrow$ *a eats b*, then:
  $b\,R^{-1}\,a \Leftrightarrow$ *b is eaten by a*. (Passive voice.)

Relations on a Set

- A (binary) relation from a set A to itself is called a relation *on* the set A.
- *E.g.*, the “<” relation from earlier was defined as a relation *on* the set **N** of natural numbers.
- The *identity relation* $\mathbf{I}_A$ on a set A is the set $\{(a,a)|a\in A\}$.

Reflexivity

- A relation R on A is *reflexive* if $\forall a \in A$, aRa.
  - *E.g.*, the relation $\geq :\equiv \{(a,b) \mid a \geq b\}$ is reflexive.
- A relation is *irreflexive* iff its complementary relation is reflexive.
  - Note "*irreflexive*" ≠ "*not reflexive*"!
  - Example: < is irreflexive.
  - Note: "likes" between people is not reflexive, but not irreflexive either. (Not everyone likes themselves, but not everyone dislikes themselves either.)

Symmetry & Antisymmetry

- A binary relation R on A is symmetric iff $R = R^{-1}$, that is, if $(a,b)\in R \leftrightarrow (b,a)\in R$.
  - *E.g.*, = (equality) is symmetric. < is not.
  - "is married to" is symmetric, "likes" is not.
- A binary relation R is *antisymmetric* if $(a,b)\in R \rightarrow (b,a)\notin R$.
  - < is antisymmetric, "likes" is not.

Transitivity

- A relation R is *transitive* iff (for all a,b,c)

  $$(a,b)\in R \land (b,c)\in R \rightarrow (a,c)\in R.$$

- A relation is *intransitive* if it is not transitive.
- Examples: “is an ancestor of” is transitive.
- “likes” is intransitive.
- “is within 1 mile of” is… ?

Composite Relations

- Let R:$A \leftrightarrow B$, and S:$B \leftrightarrow C$. Then the *composite* $S \circ R$ of R and S is defined as:

$$S \circ R = \{(a,c) \mid \exists b: aRb \wedge bSc\}$$

- Note function composition $f \circ g$ is an example.
- The n^{th} power R^n of a relation R on a set A can be defined recursively by:

$$R^0 :\equiv \mathbf{I}_A; \qquad R^{n+1} :\equiv R^n \circ R \quad \text{for all } n \geq 0.$$

  - Negative powers of R can also be defined if desired, by $R^{-n} :\equiv (R^{-1})^n$.

n-ary Relations

- An n-ary relation R on sets $A_1,\ldots,A_n$, written $R{:}A_1,\ldots,A_n$, is a subset $R \subseteq A_1 \times \ldots \times A_n$.
- The sets A_i are called the *domains* of R.
- The *degree* of R is n.
- R is *functional in domain* A_i if it contains at most one n-tuple $(\ldots, a_i,\ldots)$ for any value a_i within domain A_i.

Representing Relations

- Some ways to represent *n*-ary relations:
  - With an explicit list or table of its tuples.
  - With a function from the domain to {**T**,**F**}.
    - Or with an algorithm for computing this function.
- Some special ways to represent binary relations:
  - With a zero-one matrix.
  - With a directed graph.

Using Zero-One Matrices

- To represent a relation R by a matrix $\mathbf{M}_R = [m_{ij}]$, let $m_{ij} = 1$ if $(a_i,b_j)\in R$, else 0.
- *E.g.*, Joe likes Susan and Mary, Fred likes Mary, and Mark likes Sally.
- The 0-1 matrix representation of that "Likes" relation:

$$\begin{array}{r c} & \begin{array}{ccc} \text{Susan} & \text{Mary} & \text{Sally} \end{array} \\ \begin{array}{r} \text{Joe} \\ \text{Fred} \\ \text{Mark} \end{array} & \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array}$$

Zero-One Reflexive, Symmetric

- Terms: *Reflexive*, *non-Reflexive*, *irreflexive*, s*ymmetric, asymmetric, and antisymmetric*.
  - These relation characteristics are very easy to recognize by inspection of the zero-one matrix.

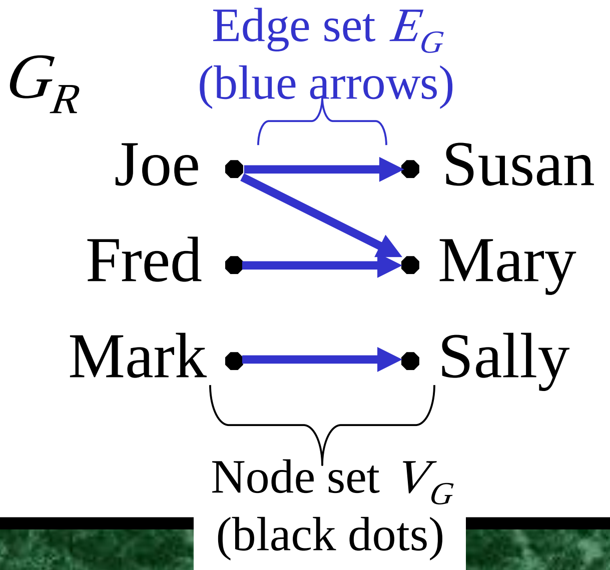
$$\begin{bmatrix} 1 & & & \text{any-thing} \\ & 1 & & \\ & & 1 & \\ \text{any-thing} & & & 1 \end{bmatrix} \quad \begin{bmatrix} 0 & & & \text{any-thing} \\ & 0 & & \\ & & 0 & \\ \text{any-thing} & & & 0 \end{bmatrix} \quad \begin{bmatrix} & 1 & & \\ 1 & \text{anything} & & 0 \\ & & & \\ & 0 & & \end{bmatrix} \quad \begin{bmatrix} & 0 & & \\ 1 & \text{anything} & 0 & 1 \\ & 0 & & \\ & 0 & & \end{bmatrix}$$

Reflexive: all 1's on diagonal	*Irreflexive*: all 0's on diagonal	*Symmetric*: all identical across diagonal	*Antisymmetric*: all 1's are across from 0's

Using Directed Graphs

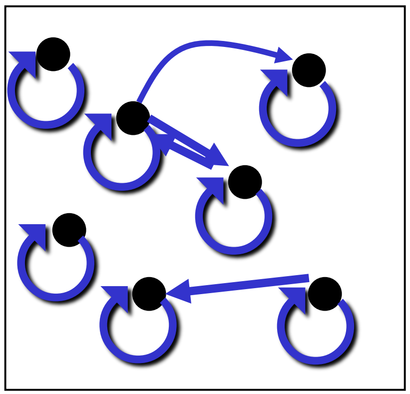
- A *directed graph* or *digraph* $G=(V_G,E_G)$ is a set V_G of *vertices (nodes)* with a set $E_G \subseteq V_G \times V_G$ of *edges (arcs,links)*. Visually represented using dots for nodes, and arrows for edges. Notice that a relation $R{:}A \leftrightarrow B$ can be represented as a graph $G_R=(V_G=A \cup B,\ E_G=R)$.

$$\mathbf{M}_R \quad \begin{array}{r} \\ \text{Joe} \\ \text{Fred} \\ \text{Mark} \end{array} \begin{array}{c} \begin{array}{ccc} \text{Susan} & \text{Mary} & \text{Sally} \end{array} \\ \left[\begin{array}{ccc} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}\right] \end{array}$$

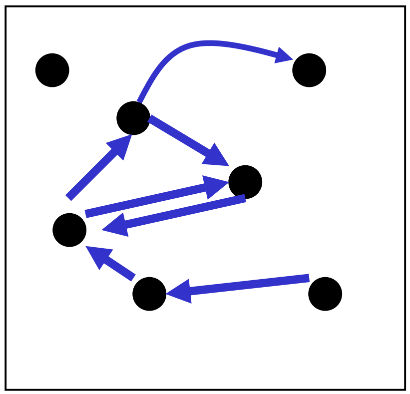


Digraph Reflexive, Symmetric

It is extremely easy to recognize the reflexive/irreflexive/ symmetric/antisymmetric properties by graph inspection.

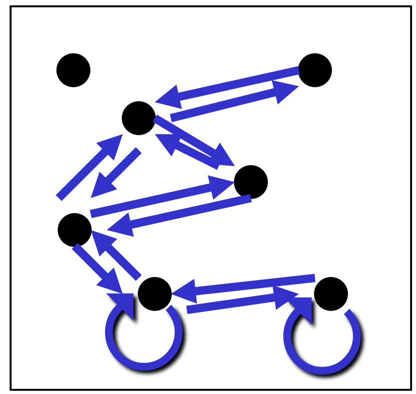


Reflexive:
Every node has a self-loop

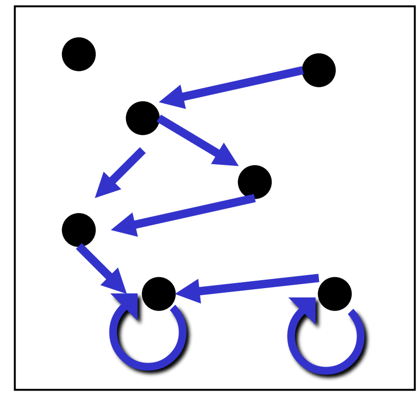


Irreflexive:
No node links to itself

Asymmetric, non-antisymmetric



Symmetric:
Every link is bidirectional



Antisymmetric:
No link is bidirectional

Non-reflexive, non-irreflexive

Closures of Relations

- For any property X, the "X closure" of a set A is defined as the "smallest" superset of A that has the given property.
- The *reflexive closure* of a relation R on A is obtained by adding (a,a) to R for each $a \in A$. *I.e.*, it is $\boxed{R \cup I_A}$
- The *symmetric closure* of R is obtained by adding (b,a) to R for each (a,b) in R. *I.e.*, it is $\boxed{R \cup R^{-1}}$
- The *transitive closure* or *connectivity relation* of R is obtained by repeatedly adding (a,c) to R for each $(a,b),(b,c)$ in R.
  - *I.e.*, it is $$\boxed{R^* = \bigcup_{n \in \mathbf{Z}^+} R^n}$$

Paths in Digraphs/Binary Relations

- A *path* of length n from node a to b in the directed graph G (or the binary relation R) is a sequence $(a,x_1), (x_1,x_2), \ldots, (x_{n-1},b)$ of n ordered pairs in E_G (or R).
  - An empty sequence of edges is considered a path of length 0 from a to a.
  - If any path from a to b exists, then we say that a is *connected to b*. ("You can get there from here.")
- A path of length $n \geq 1$ from a to a is called a *circuit* or a cycle.
- Note that there exists a path of length n from a to b in R if and only if $(a,b) \in R^n$.

Equivalence Relations

- An *equivalence relation* (e.r.) on a set A is simply any binary relation on A that is reflexive, symmetric, and transitive.
  – *E.g.*, = itself is an equivalence relation.
  – For any function $f:A \rightarrow B$, the relation "have the same f value", or $=_f :\equiv \{(a_1,a_2) \mid f(a_1)=f(a_2)\}$ is an equivalence relation, *e.g.*, let m="mother of" then $=_m$ = "have the same mother" is an e.r.

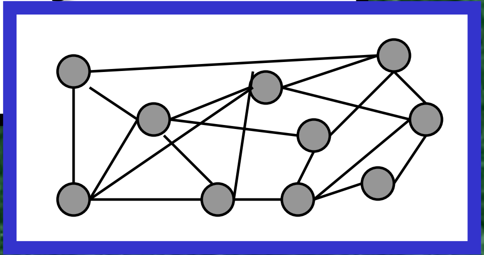
Equivalence Relation Examples

- "Strings *a* and *b* are the same length."
- "Integers *a* and *b* have the same absolute value."
- "Real numbers *a* and *b* have the same fractional part (*i.e.*, $a - b \in \mathbf{Z}$)."
- "Integers *a* and *b* have the same residue modulo *m*." (for a given $m>1$)

Graph Theory

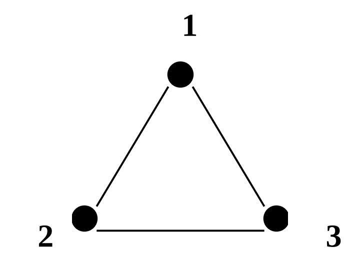
9.1: What are Graphs?

- General meaning in everyday math:
  A plot or chart of numerical data using a coordinate system.
- Technical meaning in discrete mathematics:
  A particular class of discrete structures (to be defined) that is useful for representing relations and has a convenient webby-looking graphical representation.



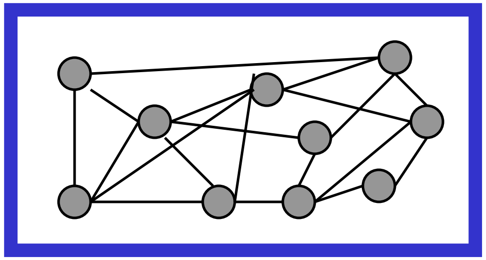
Applications of Graphs

- Potentially anything (graphs can represent relations, relations can describe the extension of any predicate).
- Apps in networking, scheduling, flow optimization, circuit design, path planning.
- Geneology analysis, computer game-playing, program compilation, object-oriented design, …



Types of Graphs: 1. Simple Graphs

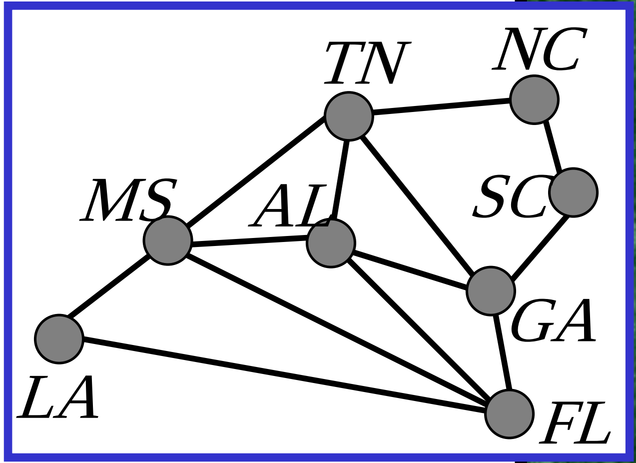
- Correspond to symmetric binary relations *R*.
- A *simple graph G*=(*V*,*E*) consists of:
  - a set *V* of *vertices* or *nodes* (*V* corresponds to the universe of the relation *R*),
  - a set *E* of *edges* / *arcs* / *links*: unordered pairs of [distinct?] elements $u,v \in V$, such that *uRv*.



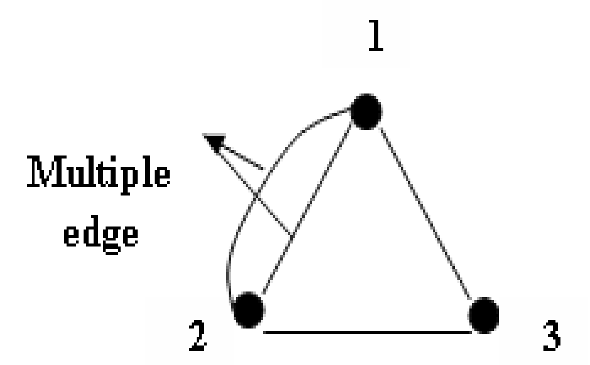
Visual Representation of a Simple Graph

Example of a *Simple Graph*

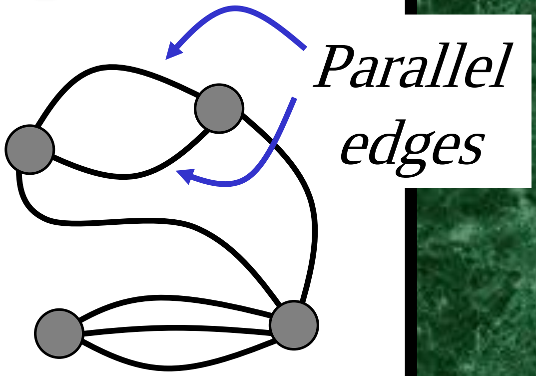
- Let V be the set of states in the far-southeastern U.S.:
  - V={FL, GA, AL, MS, LA, SC, TN, NC}
- Let $E=\{\{u,v\}|u$ adjoins $v\}$
  ={{FL,GA},{FL,AL},{FL,MS}, {FL,LA},{GA,AL},{AL,MS}, {MS,LA},{GA,SC},{GA,TN}, {SC,NC},{NC,TN},{MS,TN}, {MS,AL}}



2. Multigraphs

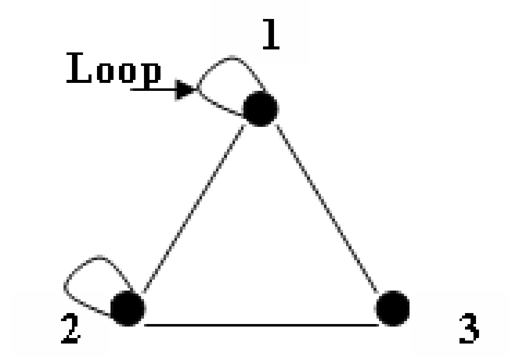


- Like simple graphs, but there may be *more than one* edge connecting two given nodes.
- A *multigraph* $G=(V, E, f)$ consists of a set V of vertices, a set E of edges (as primitive objects), and a function $f{:}E\rightarrow\{\{u,v\}|u,v\in V \wedge u\neq v\}$.

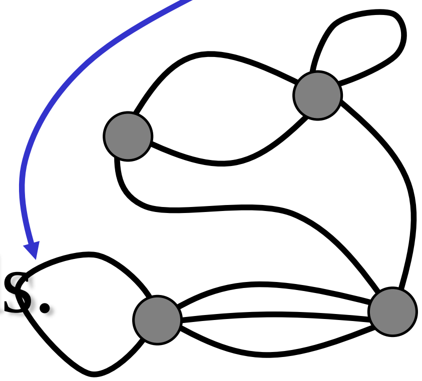


- E.g., nodes are cities, edges are segments of major highways.

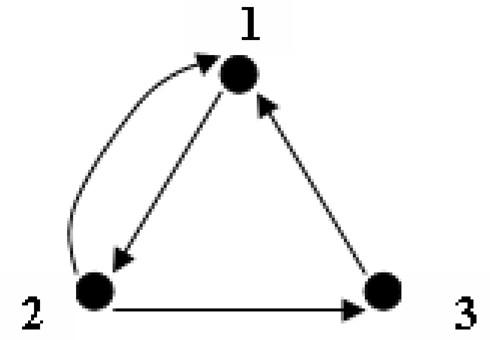
3. Pseudographs



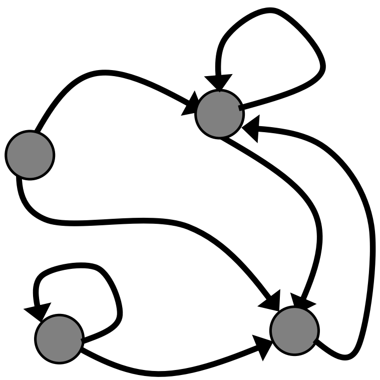
- Like a multigraph, but edges connecting a node to itself are allowed.
- A *pseudograph* $G=(V, E, f)$ where $f{:}E\rightarrow\{\{u,v\}|u,v\in V\}$. Edge $e\in E$ is a *loop* if $f(e)=\{u,u\}=\{u\}$.
- *E.g.*, nodes are campsites in a state park, edges are hiking trails through the woods.



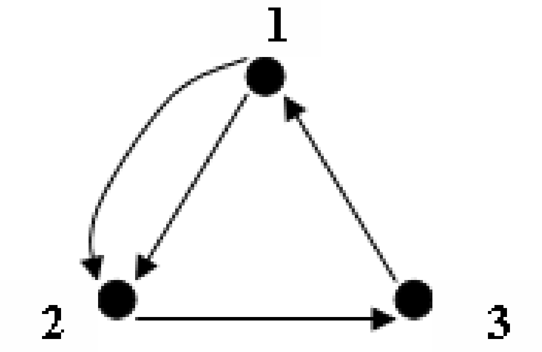
Directed Graphs



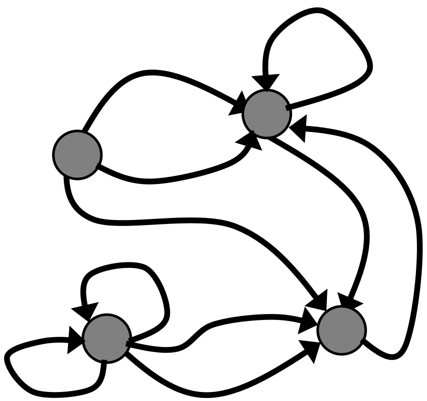
- Correspond to arbitrary binary relations *R*, which need not be symmetric.
- A *directed graph* (*V*,*E*) consists of a set of vertices *V* and a binary relation *E* on *V*.
- *E.g.*: *V* = people,
  E={(*x*,*y*) | *x* loves *y*}



Directed Multigraphs



- Like directed graphs, but there may be more than one arc from a node to another.
- A *directed multigraph* $G=(V, E, f)$ consists of a set V of vertices, a set E of edges, and a function $f:E\rightarrow V\times V$.
- E.g., V=web pages, E=hyperlinks. *The WWW is a directed multigraph...*



Types of Graphs: Summary

- Summary of the book's definitions.
- Keep in mind this terminology is not fully standardized...

Term	Edge type	Multiple edges ok?	Self-loops ok?
Simple graph	Undir.	No	No
Multigraph	Undir.	Yes	No
Pseudograph	Undir.	Yes	Yes
Directed graph	Directed	No	Yes
Directed multigraph	Directed	Yes	Yes

9.2: Graph Terminology

- *Adjacent, connects, endpoints, degree, initial, terminal, in-degree, out-degree, complete, cycles, wheels, n-cubes, bipartite, subgraph, union.*

Adjacency

Let G be an undirected graph with edge set E. Let $e \in E$ be (or map to) the pair $\{u,v\}$. Then we say:

- u, v are *adjacent* / *neighbors* / *connected*.
- Edge e is *incident with* vertices u and v.
- Edge e *connects* u and v.
- Vertices u and v are *endpoints* of edge e.

Degree of a Vertex

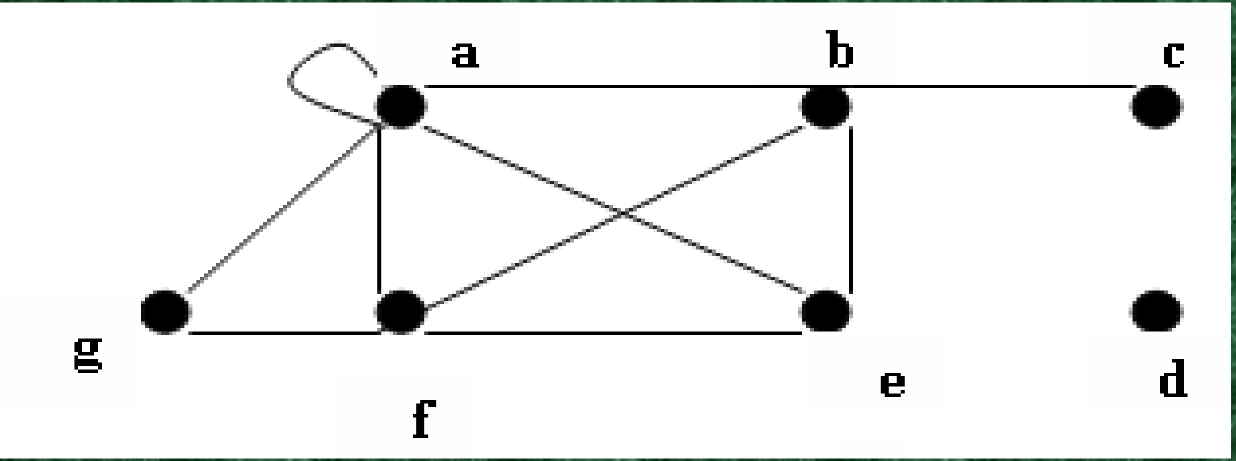
- Let G be an undirected graph, $v \in V$ a vertex.
- The *degree* of v, $\deg(v)$, is its number of incident edges. (Except that any self-loops are counted twice.)
- A vertex with degree 0 is *isolated*.
- A vertex of degree 1 is *pendant*.

Handshaking Theorem

- Let G be an undirected (simple, multi-, or pseudo-) graph with vertex set V and edge set E. Then

$$\sum_{v \in V} \deg(v) = 2|E|$$

- Corollary: Any undirected graph has an even number of vertices of odd degree.



- **deg(a)** = 6
- **deg(b)** = 4
- **deg(c)** = 1 **pendant**
- **deg(d)** = 0 **isolated**
- **deg(e) = 3**
- **deg(f) = 4**
- **deg(g) = 2**
- **$\sum$deg(v) = 20 = 2$\sum$edges=2 x 10**

Directed Adjacency

- Let *G* be a directed (possibly multi-) graph, and let *e* be an edge of *G* that is (or maps to) (*u*,*v*). Then we say:
  - *u* is *adjacent to v*, *v* is *adjacent from u*
  - *e comes from* u, e *goes to* v.
  - *e connects u to v*, *e goes from u to v*
  - the *initial vertex* of *e* is *u*
  - the *terminal vertex* of *e* is *v*

Directed Degree

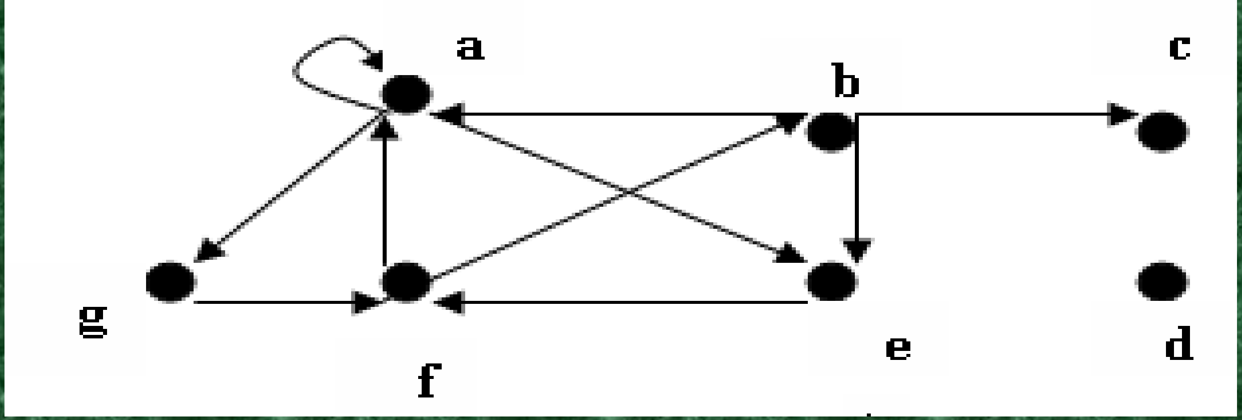
- Let G be a directed graph, v a vertex of G.
  - The *in-degree* of v, $\deg^{-}(v)$, is the number of edges going to v.
  - The *out-degree* of v, $\deg^{+}(v)$, is the number of edges coming from v.
  - The *degree* of v, $\deg(v) \equiv \deg^{-}(v) + \deg^{+}(v)$, is the sum of v's in-degree and out-degree.

Directed Handshaking Theorem

- Let G be a directed (possibly multi-) graph with vertex set V and edge set E. Then:

$$\sum_{v\in V}\deg^{-}(v)=\sum_{v\in V}\deg^{+}(v)=\frac{1}{2}\sum_{v\in V}\deg(v)=|E|$$

- Note that the degree of a node is unchanged by whether we consider its edges to be directed or undirected.



- deg+(a) = 3 deg -(a) = 3
- deg+(b) = 3 deg -(b) = 1
- deg+(c) = 0 deg -(c) = 1
- deg+(d) = 0 deg -(d) = 0
- deg+(e) = 1 deg-(e) = 2
- deg+(f) = 2 deg-(f) = 2
- deg+(g) = 1 deg-(g) = 1
- ∑deg+(v) = ∑deg-(v) =1/2 ∑deg(v)= ∑edges = 10

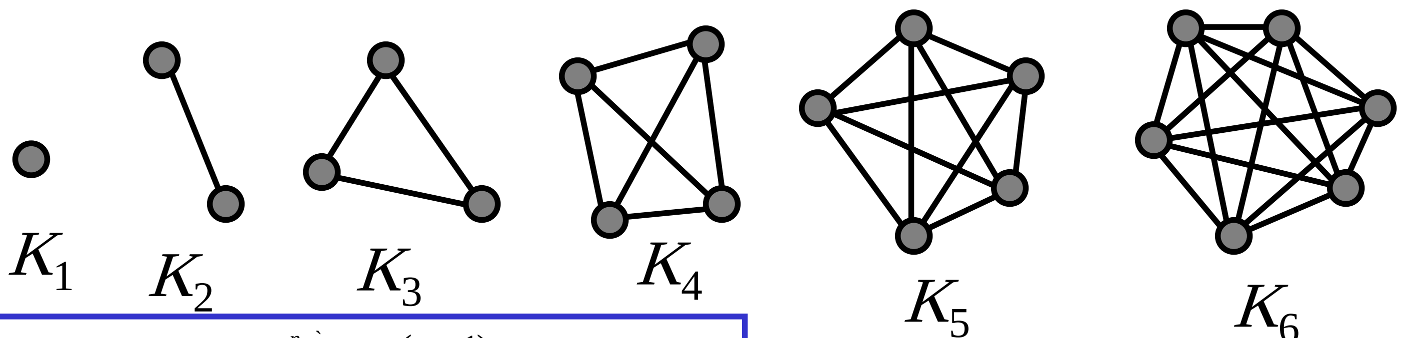
Special Graph Structures

Special cases of undirected graph structures:

- Complete graphs K_n
- Cycles C_n
- Wheels W_n
- n-Cubes Q_n
- Bipartite graphs
- Complete bipartite graphs $K_{m,n}$

Complete Graphs

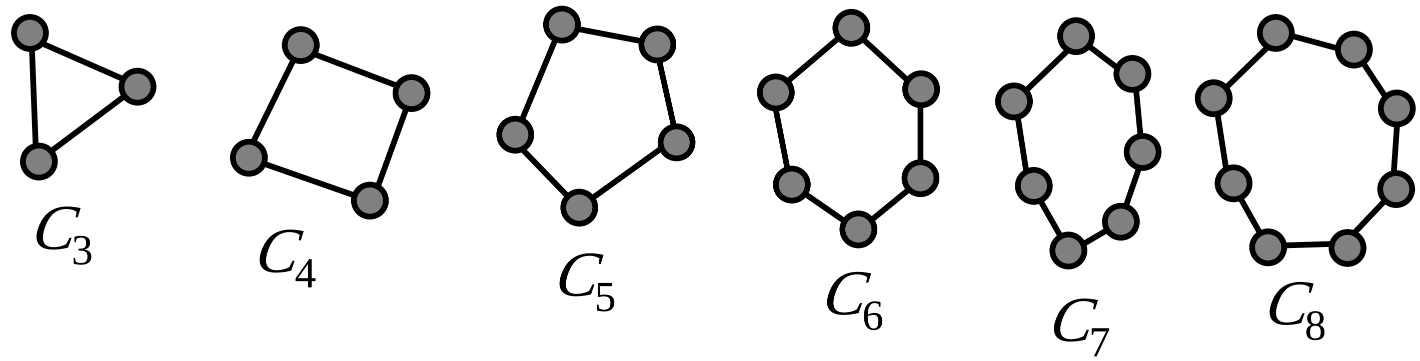
- For any $n \in \mathbf{N}$, a *complete graph* on n vertices, K_n, is a simple graph with n nodes in which every node is adjacent to every other node: $\forall u,v \in V$: $u \neq v \leftrightarrow \{u,v\} \in E$.



Note that K_n has $\sum_{i=1}^{n-1} i = \frac{n(n-1)}{2}$ edges.

Cycles

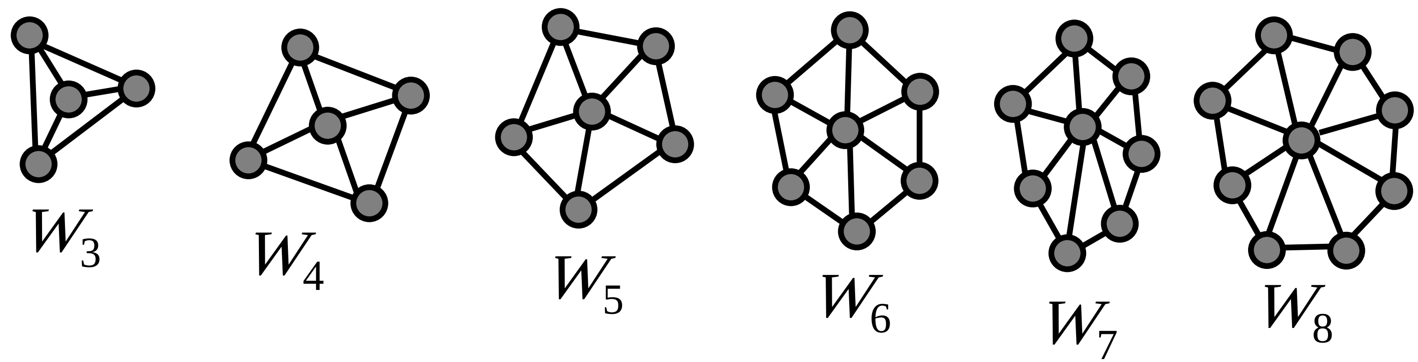
- For any $n \geq 3$, a *cycle* on n vertices, C_n, is a simple graph where $V = \{v_1, v_2, \ldots, v_n\}$ and $E = \{\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{n-1}, v_n\}, \{v_n, v_1\}\}$.



How many edges are there in C_n?

Wheels

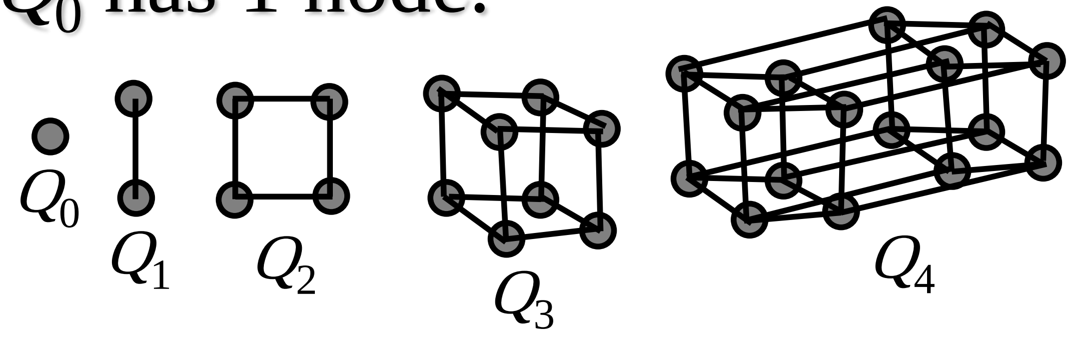
- For any $n \geq 3$, a *wheel* W_n, is a simple graph obtained by taking the cycle C_n and adding one extra vertex v_{hub} and n extra edges $\{\{v_{\text{hub}}, v_1\}, \{v_{\text{hub}}, v_2\}, \ldots, \{v_{\text{hub}}, v_n\}\}$.



How many edges are there in W_n?

n-cubes (hypercubes)

- For any $n \in \mathbf{N}$, the hypercube Q_n is a simple graph consisting of two copies of Q_{n-1} connected together at corresponding nodes. Q_0 has 1 node.



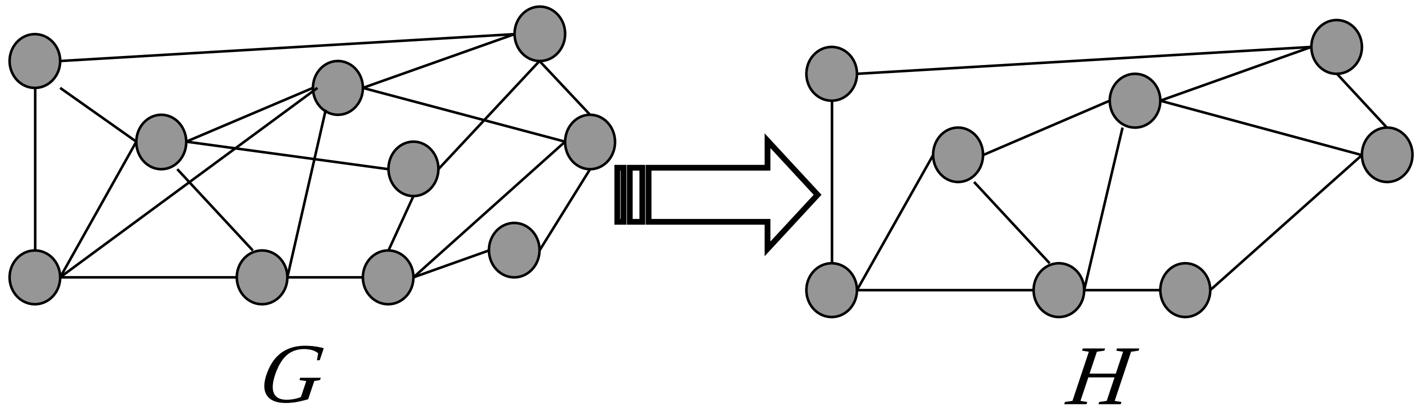
Number of vertices: 2^n. *Number of edges: Exercise to try!*

n-cubes (hypercubes)

- For any $n \in \mathbf{N}$, the hypercube Q_n can be defined recursively as follows:
  - $Q_0=\{\{v_0\},\varnothing\}$ (one node and no edges)
  - For any $n \in \mathbf{N}$, if $Q_n=(V,E)$, where $V=\{v_1,\ldots,v_a\}$ and $E=\{e_1,\ldots,e_b\}$, then $Q_{n+1}=(V\cup\{v_1',\ldots,v_a'\},$ $E\cup\{e_1',\ldots,e_b'\}\cup\{\{v_1,v_1'\},\{v_2,v_2'\},\ldots,$ $\{v_a,v_a'\}\})$ where $v_1',\ldots,v_a'$ are new vertices, and where if $e_i=\{v_j,v_k\}$ then $e_i'=\{v_j',v_k'\}$.

Subgraphs

- A subgraph of a graph $G=(V,E)$ is a graph $H=(W,F)$ where $W \subseteq V$ and $F \subseteq E$.



Graph Unions

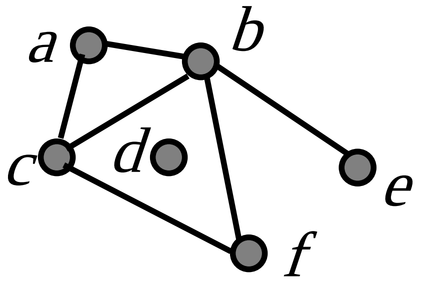
- The *union* $G_1 \cup G_2$ of two simple graphs $G_1=(V_1, E_1)$ and $G_2=(V_2,E_2)$ is the simple graph $(V_1 \cup V_2, E_1 \cup E_2)$.

9.3: Graph Representations

- Graph representations:
  - Adjacency lists.
  - Adjacency matrices.
  - Incidence matrices.

Adjacency Lists

- A table with 1 row per vertex, listing its adjacent vertices.



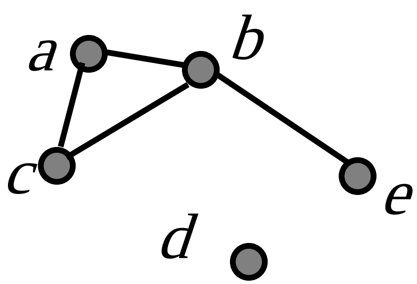
Vertex	*Adjacent Vertices*
a	*b*, *c*
b	*a*, *c*, *e*, *f*
c	*a*, *b*, *f*
d	
e	*b*
f	*c*, *b*

Directed Adjacency Lists

- 1 row per node, listing the terminal nodes of each edge incident from that node.

Adjacency Matrices

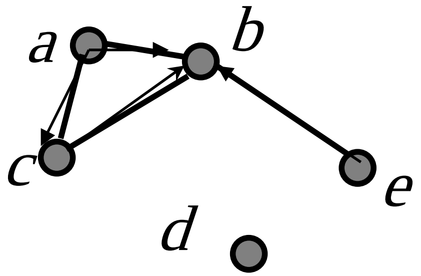
- Matrix $\mathbf{A}=[a_{ij}]$, where a_{ij} is 1 if $\{v_i, v_j\}$ is an edge of G, 0 otherwise.



$$\begin{array}{c c}
 & \begin{array}{ccccc} a & b & c & d & e \end{array} \\
\begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} & \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}
\end{array}$$

Adjacency Matrices

- Matrix $\mathbf{A}=[a_{ij}]$, where a_{ij} is 1 if $\{v_i, v_j\}$ is an edge of G, 0 otherwise.



$$\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

§8.4: Connectivity

- In an undirected graph, a *path of length n from u to v* is a sequence of adjacent edges going from vertex *u* to vertex *v*.
- A path is a *circuit* if *u*=*v*.
- A path *traverses* the vertices along it.
- A path is *simple* if it contains no edge more than once.

Paths in Directed Graphs

- Same as in undirected graphs, but the path must go in the direction of the arrows.

§9.1: Introduction to Trees

- A *tree* is a connected undirected graph with no simple circuits.
  - **Theorem:** There is a unique simple path between any two of its nodes.
- An undirected graph without simple circuits is called a *forest*.
  - You can think of it as a set of trees having disjoint sets of nodes.

Rooted Trees

- A *rooted tree* is a tree in which one node has been designated the *root*.
  - Every edge is (implicitly or explicitly) directed away from the root.
- You should know the following terms about rooted trees:
  - Parent, child, siblings, ancestors, descendents, leaf, internal node, subtree.

n-ary trees

- A rooted tree is called *n*-ary if every internal vertex has no more than *n* children.
- It is *full* if every internal vertex has *exactly* *n* children.
- A 2-ary tree is called a *binary tree*.

Ordered Rooted Tree

- A rooted tree where the children of each internal node are ordered.
- In ordered binary trees, we can define:
  - left child, right child
  - left subtree, right subtree
- For *n*-ary trees with $n>2$, can use terms like "leftmost", "rightmost," etc.

Trees as Models

- Can use trees to model the following:
  - Saturated hydrocarbons
  - Organizational structures
  - Computer file systems
- In each case, would you use a rooted or a non-rooted tree?

Some Tree Theorems

- A tree with n nodes has $n-1$ edges.
- A full m-ary tree with i internal nodes has $n=mi+1$ nodes, and $\ell=(m-1)i+1$ leaves.
  - Proof: There are mi children of internal nodes, plus the root. And, $\ell = n-i = (m-1)i+1$. □
  - Thus, given m, we can compute any of i, n, and ℓ from any of the others.

More Theorems

- **Definition:** The *level* of a node is the length of the simple path from the root to the node.
  - The *height* of a tree is maximum node level.
  - A rooted m-ary tree with height h is *balanced* if all leaves are at levels h or $h-1$.
- **Theorem:** There are at most m^h leaves in an m-ary tree of height h.
  - **Corollary:** An m-ary tree with ℓ leaves has height $h \geq \lceil \log_m \ell \rceil$. If m is full and balanced then $h = \lceil \log_m \ell \rceil$

§9.2: Applications of Trees

- Binary search trees
- Decision trees
  - Minimum comparisons in sorting algorithms
- Prefix codes
  - Huffman coding
- Game trees

§9.3: Tree Traversal

- Universal address systems
- Traversal algorithms
  - Depth-first traversal:
    - Preorder traversal
    - Inorder traversal
    - Postorder traversal
  - Breadth-first traversal
- Infix/prefix/postfix notation