[Introduction]

Discrete Structures (CSc 511)

Samujjwal Bhandari
Central Department of Computer Science and Information Technology (CDCSIT)
Tribhuvan University, Kirtipur,
Kathmandu, Nepal.

Introduction

Discrete Mathematics deals with discrete objects. Discrete objects are those objects that can be counted and are not connected for e.g. houses, trees, desks, integers, etc. So dealing with these discrete objects requires different concepts like counting techniques, knowledge of different discrete structures that are needed to understand what exactly discrete structure is like sets, relations, graphs, etc. we start our quest with foundation and go in depth later.

Logic

Logic is a language for reasoning. Since logic can helps us to reason the mathematical models it needs some rules associated with logic so that we can apply those rules for mathematical reasoning. There are lots of applications of logic in the field of computer science for e.g. designing circuits, programming, program verifications, etc.

Propositions and Propositional Calculus

Proposition is a fundamental concept in logic. Proposition is a declarative sentence that is either true or false, but not both. See the examples below:

2 + 2 = 5. (False), 7 - 1 = 6. (True)

It is hot today. (If it is hot then true)

Kathmandu is the capital of Nepal. (True)

All the above examples are either true or false.

Try to analyze the sentences below:

x > 5, Come here, Who are you?, 3 + 4

The above sentences are not propositions since we cannot say whether they are true or false.

Propositions are denoted conventionally by using small letters like p, q, r, s …. The truth value of proposition is denoted by **T** for true proposition and **F** for false proposition.

Reminder: p, q ,r ,s … are not actual propositions but they are propositional variables i.e. place holders for propositions.

The logic that deals with propositions is called propositional logic or propositional calculus.

Logical Operators/Connectives

Logical operators are used to construct mathematical statements having one or more propositions by combining the propositions. The combined proposition is called compound Proposition. The truth table is used to get the relationship between truth values of propositions. Here we present the logical operators along with their behavior in truth table:

Negation (not)

Given a proposition p, negation operator (¬) is used to get negation of p denoted by ¬p called "not p".

Example: Negation of the proposition " I love birds" is " I do not love birds" if the sentence I love birds is denoted by p then its negation is denoted by ¬p.

Truth table

p	¬p
T	**F**
F	**T**

Conjunction (and)

Given two propositions p and q, the proposition "p and q" denoted by p∧q is the proposition that is true whenever both the propositions p and q are true, false otherwise. The proposition that is obtained by the use of "and" operator is also called conjunction of p and q.

Example: If we have propositions p = "Ram is intelligent" and q = "Ram is diligent" the conjunction of p and q is Ram is intelligent and diligent. This proposition is true only when Ram is intelligent and he is diligent also, false otherwise.

Truth Table

p	q	p∧q
T	**T**	**T**
T	**F**	**F**
F	**T**	**F**
F	**F**	**F**

Disjunction (or)

Given two propositions p and q, the proposition "p or q" denoted by $p \vee q$ is the proposition that is false whenever both the propositions p and q are false, true otherwise. The proposition that is obtained by the use of "or" operator is also called disjunction of p and q.

Example: If we have propositions p = "Ram is intelligent" and q = "Ram is diligent" the disjunction of p and q is Ram is intelligent or he is diligent. This proposition is false only when Ram is not intelligent and not diligent, true otherwise.

Truth Table

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

Exclusive or (Xor)

Given two propositions p and q, the proposition exclusive or of p and q denoted by $p \oplus q$ is the proposition that is true whenever only one of the propositions p and q is true, false otherwise. As opposed to the disjunction above which is inclusive the general meaning of the English sentence can be used to know whether the "or" used is inclusive or exclusive.

Example: If we have propositions p = "Ram drinks coffee in the morning" and q = "Ram drinks tea in the morning" the exclusive or of p and q is Ram drinks coffee or tea in the morning.

Truth Table

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

Implication

Given two propositions p and q, the proposition implication $p \rightarrow q$ is the proposition that is false when p is true and q is false, true otherwise. Here p is called "hypothesis" or "antecedent" or "premise" and q is called "conclusion" or "consequence".

We come across the implication in many places in mathematical reasoning and we use different terminologies to express $p \rightarrow q$ like: "if p, then q", "q is consequence of p", "p is sufficient for q", "q if p" "q is necessary for p", "q follows from p", "if p, q", "p implies q", "p only if q","q whenever p","q provides p"

Example: p = "today is Sunday" q = "it is hot" then the implication can be "if today is Sunday then it is hot today" or "today is Sunday only if it is hot today".

Truth Table

p	q	$p \rightarrow q$
T	**T**	**F**
T	**F**	**F**
F	**T**	**T**
F	**F**	**T**

Contrapositive, Inverse and Converse

Some of the related implications formed from $p \rightarrow q$ are:

Converse: $q \rightarrow p$ (if it is hot today then today is Sunday).

Inverse: $\neg p \rightarrow \neg q$ (if today is not Sunday then it is not hot today).

Contrapositive: $\neg q \rightarrow \neg p$ (if it is not hot then today is not Sunday).

(Is contrapositive same as $p \rightarrow q$? verify!!!).

Biconditional

Given propositions p and q, the biconditional $p \leftrightarrow q$ is a proposition that is when p and q have same truth values. Alternatively $p \leftrightarrow q$ is true whenever both $q \rightarrow p$ and $q \rightarrow p$ are true. Some of the terminologies used for biconditional are:

"p if and only if q" "if p then q, and conversely" "p is necessary and sufficient for q"

Example: For propositions given above in implication, "today is Sunday if and only if it is hot today".

Truth Table

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

More Examples logical connectives:

1) Let, p = "it rained last night"

q = "the sprinkles came on last night"

r = "the lawn was wet this morning"

Translate the following into English $\neg p$, $r \wedge \neg p$ and $\neg r \vee p \vee q$.

$\neg p$ = " it didn't rain last night"

$r \wedge \neg p$ = "the lawn was wet this morning and it didn't rain last night"

$\neg r \vee p \vee q$ = "either the lawn was not wet this morning or it rained last night or the sprinkles came on last night"

2) Let p, q and r be the propositions with truth values **T**, **F**, **T** respectively. Evaluate the following:

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

$\neg r \vee \neg(p \wedge q) = \mathbf{F} \vee \neg(\mathbf{T} \wedge \mathbf{F}) = \mathbf{F} \vee \mathbf{T} = \mathbf{T}$ (true)

$\neg(p \vee q) \wedge (\neg r \vee q) = \neg(\mathbf{T} \vee \mathbf{F}) \wedge (\mathbf{F} \vee \mathbf{F}) = \mathbf{F} \wedge \mathbf{F} = \mathbf{F}$ (false)

Note: To translate English sentences to the proposition symbolic form follow these steps:

Restate the given sentence into building block sentences.

Give the symbol to each sentence and substitute the symbols using connectives

For e.g. "if it is snowing then I will go to the beach"

Restate into "it is snowing" give it symbol p and "I will go to the beach" and give it symbol q then we can write it as $p \rightarrow q$.

Propositional Equivalences

Given two propositions that differ in their syntax we may get the exactly same semantic for both the proposition. If two propositions are semantically identical then we say those two propositions are "equivalent". Such constructs are very useful in mathematical reasoning where we can substitute such propositions to equivalent propositions to construct mathematical arguments.

Tautology and Contradiction

A compound proposition that is always true, no matter what the truth values of the atomic propositions that contain in it, is called a tautology. For e.g. $p \vee \neg p$ is always true (verify!!!).

A compound proposition that is always false is called contradiction. For e.g. $p \wedge \neg p$ is always false (verify!!!).

A compound proposition that is neither a tautology nor a contradiction is called a contingency.

Logical Equivalences

The compound propositions p and q are logically equivalent, denoted by $p \Leftrightarrow q$ or $p \equiv q$, if proposition $p \leftrightarrow q$ is a tautology.

Some important logical equivalences

$p \wedge \mathbf{T} \Leftrightarrow p$	Identity law
$p \vee \mathbf{F} \Leftrightarrow p$	Identity law
$p \wedge \mathbf{F} \Leftrightarrow \mathbf{F}$	Domination law
$p \vee \mathbf{T} \Leftrightarrow \mathbf{T}$	Domination law
$p \wedge p \Leftrightarrow p$	Idempotent law
$p \vee p \Leftrightarrow p$	Idempotent law
$\neg(\neg p) \Leftrightarrow p$	Double negation law
$p \wedge q \Leftrightarrow q \wedge p$	Commutative law
$p \vee q \Leftrightarrow q \vee p$	Commutative law
$(p \wedge q) \wedge r \Leftrightarrow p \wedge (q \wedge r)$	Associative law

$(p \vee q) \vee r \Leftrightarrow p \vee (q \vee r)$	Associative law
$p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$	Distributive law
$p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$	Distributive law
$\neg(p \wedge q) \Leftrightarrow \neg p \vee \neg q$	De Morgan's law
$\neg(p \vee q) \Leftrightarrow \neg p \wedge \neg q$	De Morgan's law
$p \wedge \neg p \Leftrightarrow \mathbf{F}$	Trivial tautology
$p \vee \neg p \Leftrightarrow \mathbf{T}$	Trivial tautology
$p \rightarrow q \Leftrightarrow \neg p \vee q$	Implication
$p \leftrightarrow q \Leftrightarrow (p \rightarrow q) \wedge (q \rightarrow p)$	Equivalence
$(p \wedge q) \rightarrow r \Leftrightarrow p \rightarrow (q \rightarrow r)$	Exportation
$(p \rightarrow q) \wedge (p \rightarrow \neg q) \Leftrightarrow \neg p$	Absurdity
$p \rightarrow q \Leftrightarrow \neg q \rightarrow \neg p$	Contrapositive
$p \wedge (p \vee q) \Leftrightarrow p$	Absorption law
$p \vee (p \wedge q) \Leftrightarrow p$	Absorption law

Proving logical equivalence

a) Truth Table (for $(p \rightarrow q) \wedge (p \rightarrow \neg q) \Leftrightarrow \neg p$)

p	¬p	q	¬q	p→q	p→¬q	p→q∧p→¬q
T	**F**	**T**	**F**	**T**	**F**	**F**
T	**F**	**F**	**T**	**F**	**T**	**F**
F	**T**	**T**	**F**	**T**	**T**	**T**
F	**T**	**F**	**T**	**T**	**T**	**T**

b) Symbolic Derivation

Prove $(p \wedge \neg q) \rightarrow \neg (p \leftrightarrow r) \Leftrightarrow \neg p \vee q \vee \neg r$

Solution:

	$(p \wedge \neg q) \rightarrow \neg (p \leftrightarrow r)$	
$\equiv$	$\neg(p \wedge \neg q) \vee \neg (p \leftrightarrow r)$	[Implication]
$\equiv$	$(\neg p \vee q) \vee \neg (p \leftrightarrow r)$	[De Morgan's law]
$\equiv$	$(\neg p \vee q) \vee \neg[(p \rightarrow r) \wedge (r \rightarrow p)]$	[Biconditional equivalence]
$\equiv$	$(\neg p \vee q) \vee \neg[(\neg p \vee r) \wedge (\neg r \vee p)]$	[Implication]

$\equiv$ $(\neg p \vee q) \vee \neg[\{(\neg p \vee r) \wedge \neg r\} \vee \{(\neg p \vee r) \wedge p\}]$ [Distributive law]

$\equiv$ $(\neg p \vee q) \vee \neg[\{(\neg r \wedge \neg p) \vee (r \wedge \neg r)\} \vee \{(p \wedge \neg p) \vee (r \wedge p)\}]$ [Distributive law]

$\equiv$ $(\neg p \vee q) \vee \neg[\{(\neg r \wedge \neg p) \vee \mathbf{F}\} \vee \{\mathbf{F} \vee (r \wedge p)\}]$ [Trivial tautology]

$\equiv$ $(\neg p \vee q) \vee \neg[(\neg r \wedge \neg p) \vee (r \wedge p)]$ [Identity law]

$\equiv$ $(\neg p \vee q) \vee \neg(\neg(r \wedge q)) \wedge \neg(r \wedge p)$ [De Morgan's law]

$\equiv$ $(q \vee \neg p) \vee ((r \vee p) \wedge \neg(r \wedge p))$ [Commutative and Double negation]

$\equiv$ $q \vee (\neg p \vee ((r \vee p) \wedge \neg(r \wedge p)))$ [Associative law]

$\equiv$ $q \vee ((\neg p \vee (r \vee p)) \wedge (\neg p \vee \neg(r \wedge p)))$ [Distributive law]

$\equiv$ $q \vee (((\neg p \vee p) \vee r) \wedge (\neg p \vee \neg(r \wedge p)))$ [Associative and commutative laws]

$\equiv$ $q \vee ((\mathbf{T} \vee r) \wedge (\neg p \vee \neg(r \wedge p)))$ [Trivial tautology]

$\equiv$ $q \vee (\mathbf{T} \wedge (\neg p \vee \neg(r \wedge p)))$ [Domination law]

$\equiv$ $q \vee (\neg p \vee \neg(r \wedge p))$ [Identity law]

$\equiv$ $q \vee (\neg p \vee (\neg r \vee \neg p))$ [De Morgan's law]

$\equiv$ $q \vee ((\neg p \vee \neg p) \vee \neg r)$ [Commutative and Associative laws]

$\equiv$ $q \vee (\neg p \vee \neg r)$ [Idempotent law]

$\equiv$ $\neg p \vee q \vee \neg r$ [Associative and Commutative laws]

Proved.

Predicate

We studied propositional logic. Lets take a statement "x > 5" is this statement a proposition? The answer is no. Whenever the statements have variable(s) in them we cannot say those statements as a proposition. The question here is can we make such statements to propositions? The answer here is yes.

In the above statement there are two parts one is the variable part called "subject" and another is relation part ">5" called "predicate". We can denote the statement "x>5" by P(x) where P is predicate ">5" and x is the variable. We also call P as a propositional function where P(x) gives value of P at x. Once value is assigned to the propositional function then we can tell whether it is true or false i.e. a proposition.

For e.g. if we put the value of x as 3 and 7 then we can conclude that P(3) is false since 3 is not greater than 5 and p(7) is true since 7 is greater than 5.

We can also denote a statements with more than one variable using predicate like for the statement "x = y" we can write P(x,y) such that P is the relation "equals to" . Similarly the statements with higher number of variables can be expressed.

Remember: The logic involving predicates is called Predicate Logic or Predicate calculus similar to logic involving propositions is Propositional Logic or Propositional Calculus

Quantifiers

Quantifiers are the tools to make the propositional function a proposition. Construction of propositions from the predicates using quantifiers is called quantification. The variables that appear in the statement can take different possible values and all the possible values that the variable can take forms a domain called "Universe of Discourse" or "Universal set". We study two types of quantifier Universal quantifier and Existential quantifier.

Universal Quantifier

Universal quantifier, denoted by $\forall$, is used for universal quantification. The universal quantification of P(x), denoted by $\forall x$ P(x), is a proposition "P(x) is true for all the values of x in the universe of discourse".

We can represent the universal quantification by using the English language like:

"for all x P(x) holds" or "for every x P(x) holds" or "for each x P(x) holds".

Example:

Take universe of discourse a set of all students of CDCSIT.

P(x) represents x takes graphics class.

Here universal quantification is $\forall x$ P(x), i.e. "all students of CDCSIT take graphics class", is a proposition.

The universal quantification is conjunction of all the propositions that are obtained by assigning the value of the variable in the predicate. Going back to above example if universe of discourse is a set {ram, shyam, hari, sita} then the truth value of the universal quantification is given by P(ram) $\wedge$ P(shyam) $\wedge$ P(hari) $\wedge$ P(sita) i.e. it is true only if all the atomic propositions are true.

Existential Quantifier

Universal quantifier, denoted by ∃, is used for existential quantification. The existential quantification of P(x), denoted by ∃x P(x), is a proposition "P(x) is true for some values of x in the universe of discourse". The other forms of representation include "there exists x such that P(x) is true" or " P(x) is true for at least one x".

Example:

For the same problem given in universal quantification ∃x P(x) is a proposition is represent like " some students of CDCSIT take graphics class".

The existential quantification is the disjunction of all the propositions that are obtained by assigning the values of the variable from the universe of discourse. So the above example is equivalent to P(ram) ∨ P(shyam) ∨ P(hari) ∨ P(sita), where all the instances of variable are as in example of universal quantification. Here if at least one of the students takes graphics class then the existential quantification results true.

Free and Bound Variables

When the variable is assigned a value or it is quantified it is called bound variable. If the variable is not bounded then it is called free variable. A part of a logical expression that is quantified is given by the scope of the quantifier. We use parenthesis to give scope of the quantifier. For e.g. $\forall x (P(x)) \rightarrow q$ is not same as $\forall x (P(x) \rightarrow q)$

Example:

P(x,y) has two free variables x and y.

P(2, y) has one bound variable 2 and one free variable y.

P(2,y) where y = 4, is bounding the variable y also.

∀x P(x) has a bound variable x.

∀x P(x,y) has one bound variable x and one free variable y.

Expression with no free variable is a proposition.

Expression with at least one free variable is a predicate only.

Order of Quantification

Order of quantification goes from the left to right. If we have a quantified proposition involving two variable (nested quantifier) then the order must be considered.

Example: Let L(x,y) denotes x loves y where universe of discourse for x, y is set of all people in the world. Translate $\forall x \exists y L(x,y)$, $\exists y \forall x\ L(x,y)$, $\exists x \forall y\ L(x,y)$, $\forall y \exists x L(x,y)$, $\forall x \forall y L(x,y)$ and $\exists x \exists y L(x,y)$ into English.

Solution:

$\forall x \exists y L(x,y)$: [for all x there is some y such that x loves y i.e. everybody loves someone. This is false when there is someone who doesn't love any one]

$\exists y \forall x\ L(x,y)$: [for some y all x love y i.e. there is a people who is loved by everyone. This is false when there is no person who is loved.]

$\exists x \forall y\ L(x,y)$: [There is some x such that x loves all y i.e. there is someone who loves all the people. This is false when all people do not love some people]

$\forall y \exists x L(x,y)$: [for all y there is x who loves y i.e. everyone has someone who loves them. When this is false? (try yourself)]

$\forall x \forall y L(x,y)$: [for all x, x loves all the y i.e. everybody loves everyone. When this is false ? (try yourself)]

$\exists x \exists y L(x,y)$: [There is some x such that he loves some y i.e. someone loves somebody. When this is false? (try yourself)]

Negation of Quantifies Expression

Let P(x) denotes x is lovely, universe of discourse for x is girls in Kathmandu. Then,

$\forall x\ P(x)$ is every girl in Kathmandu is lovely. If we want to negate it the meaning would be like there is a girl in Kathmandu who is not lovely i.e. $\exists x \neg P(x)$.

$\exists x P(x)$ is at least a girl in Kathmandu is lovely. The opposite for this (negation) would be no girls in Kathmandu are lovely. i.e. $\forall x\ \neg P(x)$.

The negation of the nested quantifier can be done by successively negating the quantifier using the above negation rule for single quantifier for e.g. $\neg\ (\forall y \exists x P(x,y))$ is

$\exists y(\neg \exists x P(x,y)) = \exists y \forall x \neg\ P(x,y)$.

Translating the Sentences into Logical Expression

Example 1

Translate "not every integer is even" where the universe of discourse is set of integers.

Solution:

Let E(x) denotes x is even. $\neg\forall xE(x)$

Example 2

Translate "if a person is female and is a parent, then this person is someone's mother" into logical expression, Universe of discourse is set of all people.

Solution:

Let F(x) denotes x is female, P(x) denotes x is a parent and M(x,y) denotes x is a mother of y. then the logical expression for above sentence is $\forall x\exists y\ (F(x) \wedge P(x) \rightarrow M(x,y))$

Example 3

Translate "everyone has exactly one best friend" into logical expression where universe of discourse is set of all people.

Solution:

Let B(x,y) denotes y is best friend of x then $\forall x\exists y(B(x,y) \wedge \forall z\ (B(x,z) \rightarrow (y = z))$ is the solution.

Sets and Set Operations

Set is a very important concept in mathematics. In computer science also we deal with set in most of the case. For e.g. if we are dealing with relations in database then they are sets ordered collection of elements, similarly we can view graph as a set. Set is a collection of zero or more objects (or elements or members), the elements need not be ordered. If we denote set by S and some element from the set by e then we say "e belongs to S" or "S contains e" or in symbol we can write $e \in S$. for e.g. V = {a, e, i, o, u} is a set of vowels and $i \in V$, if some object doesn't belong to the set we write it as " does not belong to" i.e. say $x \notin V$; C = {b, c, d, f, g, h, j, k, l, m, n, p, q, r, s, t, v, w, y, z) is a set of consonants.

Representations of Sets

Some of the ways of representing a set are:

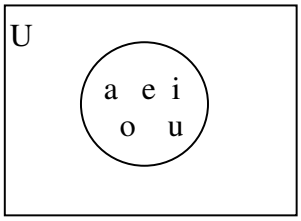
Listing of elements: Set of vowels is {a, e, i, o, u}.

Set builder form: here propertied of the members of the set is described for e.g. R = {x | x is a real number}

Recursive formula: the elements of set are defined using the previous element of the set

that is known. For e.g. set of natural numbers can be represented as $N = \{x_n = x_{n-1} + 1$, where $x_0 = 0\}$.

Venn diagram: graphical representation of set. For e.g. set of vowel as given above can be represented as:



In the above diagram the circle represents the set of vowels where as the enclosed rectangle represents the "universe of discourse" or "universe".

Some Definitions

Subset: Let A and B be two sets. Then A is said to be subset of B if every elements of A is an element of B. A is said to be proper subset of B if A is subset of B and there is at least one element in B that is not in A. Symbolically subset is represented as $A \subseteq B$ to denote that A is subset of B and $A \subset B$ to denote that A is proper subset of B.

Some subsets related properties

$A \subseteq A$; If $A \subseteq B$ and $B \subseteq C$ then $A \subseteq C$; If $A \subseteq B$ and $B \subset C$ then $A \subset C$; If $A \subseteq B$ and $A \not\subset C$ then $B \not\subset C$, where $\not\subset$ means "is not contained in". As subset is defined above **superset** can be defined in similar manner where in the above definition B is the superset of A denoted by $B \supseteq A$ for superset and $B \supset A$ for proper superset.

Equal Sets: Two sets A and B are equal if and only if they contain exactly same elements. In other words if $A \subseteq B$ and $B \subseteq A$ then $A = B$.

For e.g. $A = \{1, 2, 3, 4, 5\}$ and $B = \{2, 5, 4, 1, 3\}$ are equal sets.

Empty set: The set that contains no element is called empty set and denoted by $\varnothing$. It is also called null set. We have $\varnothing = \{\}$ but $\varnothing \neq \{\varnothing\}$.

Cardinality: For the set S, if there are exactly *n distinct* elements in S where n is a number then we say that cardinality of the set S is n denoted by $|S|$.

For e.g. $|\varnothing| = 0$; $|\{a, b, b, c, a\}| = 3$; $|\{\{a, b\}, \{a, b, c\}\}| = 3$

If $n \in N$ then the set is finite otherwise, it is infinite.

Power Set: Given a set S, power set denoted by P(S) is the set that contains all the subsets of the set S. Symbolically we can write $P(S) = \{x \mid x \subseteq S\}$. For e.g. power set for the set {2, 3} is {∅, {2}, {3}, {2,3}}. The number of elements in the power set of set having n elements is $2^{|n|}$. *Remember: ∅ is member of all power set.*

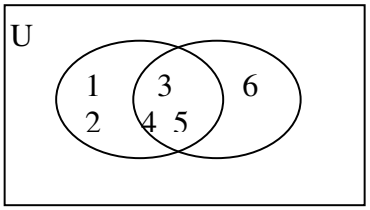
Set Operations

Union Operator: Given two sets A and B, the union of set A and set B is the set that contains those elements that are either in A or in B, or in both, denoted by $A \cup B$. Symbolically, we write union of A and B as: $A \cup B = \{x \mid x \in A \vee x \in B\}$.

Example:

$\{2, 3\} \cup \{a, b, c\} = \{2, 3, a, b, c\}$.

$\{1, 2, 3, 4, 5\} \cup \{3, 4, 5, 6, 7\} = \{1, 2, 3, 4, 5, 6, 7\}$. This can be shown in Venn diagram as

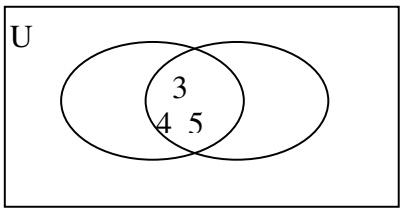


Intersection Operator: Given two sets A and B, the intersection of set A and set B is the set that contains those elements that are in both A and B, denoted by $A \cap B$. Symbolically, we write intersection of A and B as: $A \cap B = \{x \mid x \in A \wedge x \in B\}$.

Example:

$\{2, 3\} \cap \{a, b, c\} = \{\ \}$.

$\{1, 2, 3, 4, 5\} \cap \{3, 4, 5, 6, 7\} = \{3, 4, 5\}$. This can be shown in Venn diagram as



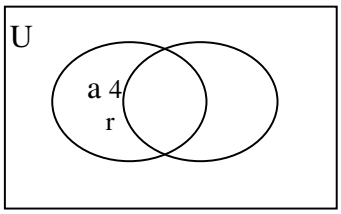
Disjoint Set: Two sets are disjoint if the intersection of two sets is null set.

Set Difference: Given two sets A and B, The difference of A and B is the set that contains all the elements that are in A but not in B, denoted by A –B. This difference is also called complement of B with respect to A. Symbolically we write difference of A

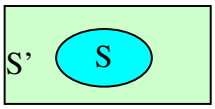
and B as $A - B = \{x \mid x \in A \wedge x \notin B\}$.

Examples:

A = {2, 4, a, r} and B = {1, 2, a, s, t} then A – B = {4, r}. Venn diagram is given below:



Complement: Complement of set S is denoted by U – S or S', where U is the universal set, is the of difference of universal set U and set S. Symbolically complement is written as $S' = \{x \mid x \notin S\}$.



In the Venn diagram shown above, the outer part from the oval is complement of S.

Set Identities

The set identities that we learn here is similar to that of propositional logic.

$A \cap U = A$	Identity law
$A \cup \varnothing = A$	Identity law
$A \cap \varnothing = \varnothing$	Domination law
$A \cup U = U$	Domination law
$A \cap A = A$	Idempotent law
$A \cup A = A$	Idempotent law
$(A')' = A$	Complementation law
$A \cap B = B \cap A$	Commutative law
$A \cup B = B \cup A$	Commutative law
$(A \cap B) \cap C = A \cap (B \cap C)$	Associative law
$(A \cup B) \cup C = A \cup (B \cup C)$	Associative law
$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$	Distributive law
$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$	Distributive law
$(A \cap B)' = A' \cup B'$	De Morgan's law
$(A \cup B)' = A' \cap B'$	De Morgan's law

Generalized Union and Intersection

Since union and intersection of the sets holds associativity we can combine the sets with same operators in any order. The union of a collection of sets is the set that contains those elements that are members of at least one set in the collection. The notation we use for this operation is $A_1 \cup A_2 \cup \ldots \cup A_n = \bigcup_{i=1}^{n} A_i$. The intersection of the collection of the sets is the set that contains those elements that are in all the sets in the collection. We represent generalized intersection as $A_1 \cap A_2 \cap \ldots \cap A_n = \bigcap_{i=1}^{n} A_i$.

Proving Set Identities

Using Mutual Subsets: Show $A \subseteq B$ and $B \subseteq A$ to show $A = B$.

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

Step1: Assume $x \in A \cap (B \cup C)$ and try to show $x \in (A \cap B) \cup (A \cap C)$.

Then we know, $x \in A$ and $x \in B$ or $x \in C$, or both.

Since $x \in A$ and $x \in B$, $x \in (A \cap B)$. Hence $x \in (A \cap B) \cup (A \cap C)$.

Since $x \in A$ and $x \in C$, $x \in (A \cap C)$. Hence $x \in (A \cap B) \cup (A \cap C)$.

Therefore, $A \cap (B \cup C) \subseteq (A \cap B) \cup (A \cap C)$.

Step 2: Assume $x \in (A \cap B) \cup (A \cap C)$ and try to show $x \in A \cap (B \cup C)$.

Then we know $x \in (A \cap B)$, that is $x \in A$ and $x \in B$, or $x \in (A \cap C)$, that is $x \in A$ and $x \in C$, or both. In any case we have $x \in A$ as true.

Since $x \in B$, $x \in (B \cup C)$. So $x \in A \cap (B \cup C)$.

Since $x \in C$, $x \in (B \cup C)$. So $x \in A \cap (B \cup C)$.

Therefore, $(A \cap B) \cup (A \cap C) \subseteq A \cap (B \cup C)$.

From step 1 and step 2 $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$.

Using Logical Equivalences: represent the set to the set builder form and apply logical equivalence transformations.

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

We can denote $(A \cup B)' = \{x \mid x \notin A \cup B\} = \{x \mid \neg(x \in A \cup B)\}$

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

$= \{x \mid x \in A' \wedge x \in B')\}$ $= \{x \mid x \in A' \cap B'\}$

$= A' \cap B'$

Membership Table: Like truth table you have studied earlier for propositional logic. Use 1 to denote set membership and 0 otherwise.

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

A	B	A - B	A - (A - B)	A ∩ B
1	1	0	1	1
1	0	1	0	0
0	1	0	0	0
0	0	0	0	0

Cartesian Products

Sets are unordered collections of objects but sometime we need ordered collections. Such ordered collections can be obtained from ordered n- tuples.

Ordered n- tuple: The ordered n tuple $(a_1, a_2, \ldots, a_n)$ is the ordered collection where a_1 is the first element, a_2 is the second element and so on …. Two ordered n – tuples are equal if and only if they have each corresponding pair of their elements is equal i.e. $(a_1, a_2, \ldots, a_n) = (b_1, b_2, \ldots, b_n)$ if and only if $a_1 = b_1, a_2 = b_2, \ldots, a_n = b_n$.

Cartesian Product: Given sets A and B. The Cartesian product of A and B is the set of all ordered pairs (a, b) where $a \in A$ and $b \in B$. The Cartesian product of A and B is denoted by $A \times B$. Symbolically, we can write it as $A \times B = \{(a, b) \mid a \in A \wedge b \in B\}$

Example: Let $A = \{a, b, c\}$ and $B = \{1, 2, 3\}$ then

$A \times B = \{(a, 1), (a, 2), (a, 3), (b, 1), (b, 2), (b, 3), (c, 1), (c, 2), (c, 3)\}$

Remember: $A \times B \neq B \times A$ unless $A = \varnothing$ or $B = \varnothing$ or $A = B$ (verify!!!)

Similarly Cartesian product of more than two sets can be defined as, Cartesian product of $A_1, A_2, \ldots, A_n$ is the set of ordered n-tuples $(a_1, a_2, \ldots, a_n)$ where each $a_i \in A_i$, for i =1, 2, …, n. It is denoted by $A_1 \times A_2 \times \ldots \times A_n$.

Symbolically, $A_1 \times A_2 \times \ldots \times A_n = \{(a1, a_2, \ldots, a_n) \mid a_i \in A_i$, for i =1, 2, …, n$\}$

Example: $A = \{1, 2\}$, $B = \{2\}$ and $C = \{a, b, c\}$ then

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

Relations (Intro)

Binary Relation: Given sets A and B, a binary relation from A to B is a subset of $A \times B$ i.e. a binary relation from A to B is a set R of ordered pairs where first element of each ordered pair belongs to set A and the second element belongs to the set B. the notation aRb is used to denote that $(a, b) \in R$ and we say a is related to b by R.

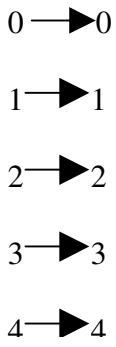
Example:

$A = \{1, 2, 3\}$ $B = \{a, b\}$ then relations from the above two sets can be

$R = \{(1, a), (2, a)\}$ $S = \{(1,a), (1, b), (2, a), (2, b)\}$ and others are also possible.

Relations on a Set: A relation on the set A is the relation from A to A i.e. it is the subset of $A \times A$.

Example: Let $A = \{0, 1, 2, 3, 4\}$ the ordered pairs that are in the relation $R = \{(a, b) \mid a = b\}$ is given by $R = \{(0, 0), (1, 1), (2, 2), (3, 3), (4, 4)\}$, We can view this relation as



R	0	1	2	3	4
0	#				
1		#			
2			#		
3				#	
4					#

Properties of Relations

Reflexive: A relation R on a set A is called reflexive if $(a, a) \in R$ for every element $a \in A$. For e.g. relation $\leq$ on set of integers is reflexive.

Symmetric: A relation R on set A is called symmetric if $(a, b) \in R$ then $(b, a) \in R$. for $a, b \in A$. For e.g. relation = on set of integers is symmetric.

Transitive: A relation R on a set A is called transitive if $(a, b) \in R$ and $(b, c) \in R$ then $(a, c) \in R$, for $a, b, c \in A$. For e.g. relation $\leq$ on set of integers is transitive.

Antisymmetric: A relation R on a set A is called antisymmetric if $(a, b) \in R$ and $(b, a) \in R$ then $a = b$, for $a, b \in A$. For e.g. relation $\geq$ on set of integers is antisymmetric.

Asymmetric: A relation R on a set A is called asymmetric if (a, b) ∈ R then (b, a) ∉ R, for a, b ∈ A. For e.g. relation > on set of integers is asymmetric.

Irreflexive: A relation R on a set A is called irreflexive if for every a ∈ A, (a, a) ∉ R. For e.g. relation > on set of integers is irreflexive.

Combining Relations

Relation from A to B is a subset of A × B so any operations that are operable in sets are also operable in relations (see notes on sets for detail).

Composite Relation: Let R and S be the relations from A to B and B to C respectively. The composite relation of R and S is a set having ordered pairs (a, c), where a∈ A and c ∈ C, and for which there exists an element b ∈ B such that (a, b) ∈ R and (b, c) ∈ . The composite relation of R and S is denoted by SoR.

Example: Let R = {(a, 1), (a, 2), (b, 1)} and S = {(1, x), (2, y)} where A = {a, b}, B ={1, 2} and C = {x, y}. Then, SoR = {(a, x), (a, y), (b, x)}

On the basis of composite relation the powers of a relation R can be defined as, The powers R^n, n =1, 2, 3, …are inductively defined as $R^1 = R$ and $R^{n+1} = R^n oR$.

Note: More on the relations will be covered later

Functions

Sometimes we assign each element form a set to the elements of other set that may be the same as the first. For e.g. each worker working on the factory to the set of the works. This kind of assignment gives rise to the function.

Function: given two A and B**,** A function f from A to B is an assignment of unique element of B to each element of A. if b is the unique element of B assigned by the function f to the element a of A the we write f(a) = b. A function from A to B is written as f: A → B. Given a function f: A → B where f(a) = b then we define following terms:

Domain: Set A is the domain of function f.

Codomain: Set B is the codomain of function f.

Image: b is the image of a.

Pre-image: a is the pre-image of b.

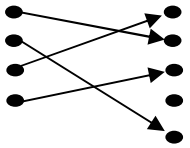
Range: Set of all images of elements of A is range.

Types of Functions

One – to – One (Injective) Function: A function f is one-to-one, if and only if f(x) = f(y) implies x = y for all x and y in the domain of f.

Example: f(x) = x2 from set of integers to the set of integers is not an injection because f(-1) = f(1) = 1 does not imply –1 = 1.

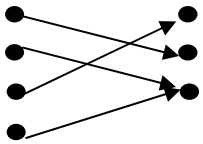
The pictorial representation of one-to-one functions looks like:



Onto (Surjective) Function: A function f is surjective or onto if and only if for every element $b \in B$ there is an element $a \in A$ such that f(a) = b.

Example: The function f(x) = x +1 from the set of integers to the set of integers is onto because for every integer b there is an integer a such that f(a) = b, where each a = b-1.

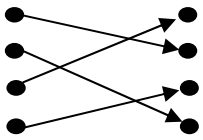
The pictorial representation of surjection is as below:



One-to-One Correspondence (Bijective Function): A function is bijection if it is both onto and one-to-one.

Example: The function f(x) = x +1 from the set of integers to the set of integers is bijection since it is one-to-one (How?) and onto (see above).

The pictorial representation of bijection is as below:



Inverse Function: given a bijective function f: A $\rightarrow$ B, the inverse of function f is denoted by f^{-1} assigns each element of B to the unique element of A such that f(a) = b. so we can write f^{-1} (b) = a when f(a) = b.

Example: The function f(x) = x +1 from the set of integers to the set of integers is bijection (see above) hence we can have inverse of it and it is denoted as f^{-1} (x) = x-1.

Growth of Functions

Complexity analysis of an algorithm is very hard if we try to analyze exact. we know that the complexity (worst, best, or average) of an algorithm is the mathematical function of the size of the input. So if we analyze the algorithm in terms of bound (upper and lower) then it would be easier i.e. understanding the growth of the function is easier. For this purpose we need the concept of asymptotic notations.

Big Oh (O) notation

When we have only asymptotic upper bound then we use O notation. A function f(x) = O(g(x)) (read as f(x) is big oh of g(x)) iff there exists two positive constants c and x_0 such that for all x >= x_0, 0 <= f(x) <= c*g(x)

The above relation says that g(x) is an upper bound of f(x)

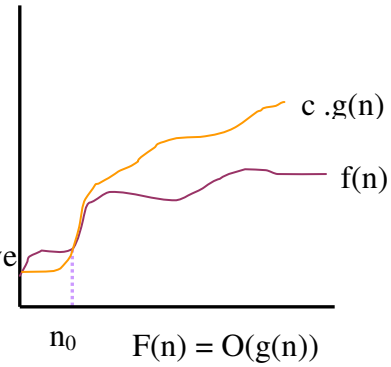
Some properties:

Transitivity : f(x) = O(g(x)) & g(x) = O(h(x)) $\Rightarrow$ f(x) = O(h(x))

Reflexivity: f(x) = O(f(x))

O(1) is used to denote constants.

For all values of n >= n_0, plot shows clearly that f(n) lies below or on the curve of c*g(n)



Examples

1. f(n) = $3n^2 + 4n + 7$

   g(n) = n^2 , then prove that f(n) = O(g(n)).

   Proof: let us choose c and n_0 values as 14 and 1 respectively then we can have f(n) <= c*g(n), n>=n0 as

   $3n^2 + 4n + 7 <= 14*n^2$ for all n >= 1

   the above inequality is trivially true

   hence f(n) = O(g(n))

2. Prove that n log (n^3) is O($\sqrt{n^3}$)).

Proof: we have n log (n^3) = 3n log n

again, $\sqrt{n^3}$ = n $\sqrt{n}$,

if we can prove log n = O($\sqrt{n}$) then problem is solved

because n log n = n O($\sqrt{n}$) that gives the question again.

We can remember the fact that log ^{a} n is O (n^b) for all a,b>0.

In our problem a = 1 and b = ½ ,

hence log n = O($\sqrt{n}$).

So by knowing log n = O($\sqrt{n}$) we proved that

n log (n^3) = O($\sqrt{n^3}$)).

3. Is 2^{n+1} =O(2^n) ?

Is 2^{2n} = O(2^n) ?

Big Omega (Ω) notation

Big omega notation gives asymptotic lower bound. A function f(x) = Ω(g(x)) (read as f(x) is big omega of g(x)) iff there exists two positive constants c and x_0 such that for all x >= x_0,

0 <= c*g(x) <= f(x).

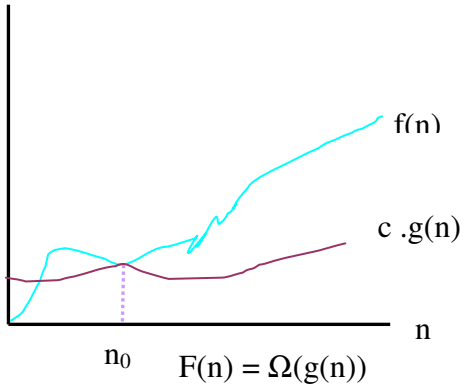
The above relation says that g(x) is an lower bound of f(x).

some properties:

Transitivity : f(x) = O(g(x)) & g(x) = O(h(x)) ⇒ f(x) = O(h(x))

Reflexivity: f(x) = O(f(x))

For all values of n >= n_0, plot shows clearly that f(n) lies above or on the curve of c*g(n).



Examples

1. f(n) = $3n^2 + 4n + 7$

g(n) = n^2 , then prove that f(n) = Ω(g(n)).

Proof: let us choose c and n_0 values as 1 and 1, respectively then we can have

f(n) >= c*g(n), n>=n0 as

$3n^2 + 4n + 7 >= 1*n^2$ for all n >= 1

the above inequality is trivially true

hence f(n) = Ω(g(n))

Big Theta (Θ) notation

When we need asymptotically tight bound then we use notation. A function f(x) = (g(x)) (read as f(x) is big theta of g(x)) iff there exists three positive constants c_1, c_2 and x_0 such that for all x >= x_0, 0 <= c_1*g(x) <= f(x) <= c_2*g(x)

The above relation says that f(x) is order of g(x)

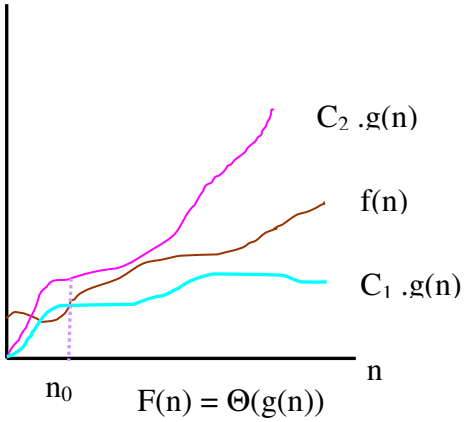
some properties:

Transitivity : f(x) = Θ (g(x)) & g(x) = Θ (h(x)) ⇒ f(x) = Θ (h(x))

Reflexivity: f(x) = Θ (f(x))

Symmetry: f(x) = Θ g(x) iff g(x) = Θ f(x)

For all values of n >= n_0, plot shows clearly that f(n) lies between c_1* g(n)and c_2*g(n).



Examples

1. f(n) = $3n^2 + 4n + 7$

   g(n) = n^2 , then prove that f(n) = (g(n)).

   Proof: let us choose c1, c2 and n_0 values as 14, 1 and 1 respectively then we can have,

   f(n) <= c1*g(n), n>=n0 as $3n^2 + 4n + 7 <= 14*n^2$, and

   f(n) >= c2*g(n), n>=n0 as $3n^2 + 4n + 7 >= 1*n^2$

   for all n >= 1(in both cases).

   So c2*g(n) <= f(n) <= c1*g(n) is trivial.

   Hence f(n) = Θ (g(n)).

2. Show $(n + a)^b = \Theta(n^b)$, for any real constants a and b, where b>0.

Here, using Binomial theorem for expanding $(n + a)^b$, we get ,

$C(b,0)n^b + C(b,1)n^{b-1}a + \ldots + C(b,b-1)na^{b-1} + C(b,b)a^b$

we can obtain some constants such that $(n + a)^b <= c_1*(n^b)$, for all $n >= n_0$

and

$(n + a)^b >= c_2*(n^b)$, for all $n >= n_0$, here we may take $c_1 = 2^b$ $c_2 = 1$ $n_0 = |a|$,

since $1 *(n^b) <= (n + a)^b <= 2^b *(n^b)$.

Hence the problem is solved.

Why $c_1 = 2^b$? since $\Sigma C(n,k) = 2^k$, where k=0 to n.

Little Oh (o) notation

Little oh (o) notation is used to denote the upper bound that is not asymptotically tight. A function f(x) = o(g(x)) (read as f(x) is little oh of g(x)) iff for any positive constant c there exists positive constant x_0 such that for all $x >= x_0$,

$0 <= f(x) < c*g(x)$

for example $4x^4$ is $O(x^4)$ but not $o(x^4)$.

Alternatively f(x) is little oh of g(x) if $\lim_{x->\infty} \frac{f(x)}{g(x)} = 0$

Note: transitivity is satisfied.

Little Omega (ω) notation

Little omega (o) notation is used to denote the lower bound that is not asymptotically tight. A function f(x) = ω(g(x)) (read as f(x) is little omega of g(x)) iff for any positive constant c there exists positive constant x_0 such that for all $x >= x_0$,

$0 <= c*g(x) < f(x)$.

for example $x^3/7$ is $\omega(x^2)$ but not $\omega(x^3)$.

Alternatively f(x) is little omega of g(x) if $\lim_{x->\infty} \frac{f(x)}{g(x)} = \infty$

Note: transitivity is satisfied.