Discrete Math for Computer Science

Peter Schaefer

Freshman Fall

Contents

1 Logic

1.1 Propositions and Logical Operations

Proposition: a statement that is either true or false.
Some examples include "It is raining today" and "$3 \cdot 8 = 20$".
However, not all statements are propositions, such as "open the door"

Name	Symbol	alternate name
NOT	$\neg$	negation
AND	$\wedge$	conjunction
OR	$\vee$	disjunction
XOR	$\oplus$	exclusive or

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

XOR is very useful for encryption and binary arithmetic.

1.2 Evaluating Compound Propositions

p : The weather is bad.	$p \wedge q$: The weather is bad *and* the trip is cancelled
q : The trip is cancelled.	$p \vee q$: The weather is bad *or* the trip is cancelled
r : The trip is delayed.	$p \wedge (q \oplus r)$: The weather is bad *and* either the trip is cancelled *or* delayed

Order of Evaluation $\neg$, then $\wedge$, then $\vee$, but parenthesis always help for clarity.

Example Truth Table:

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

1.3 Conditional Statements

$p \rightarrow q$ where p is the hypothesis and q is the conclusion

Format	Terminology
$p \rightarrow q$	given
$\neg q \rightarrow \neg p$	contrapositive
$q \rightarrow p$	converse
$\neg p \rightarrow \neg q$	inverse

given	$p \rightarrow q$	$\equiv$	$\neg q \rightarrow \neg p$	contrapositive
inverse	$\neg p \rightarrow \neg q$	$\equiv$	$q \rightarrow p$	converse

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

p is a sufficient condition for q
q is a necessary condition for p

Phrase	Logic
q if p	$p \rightarrow q$
q only if p	$q \rightarrow p$
q if and only if p	$p \leftrightarrow q$

Order of Operations: $p \wedge q \rightarrow r \equiv (p \wedge q) \rightarrow r$

1.4 Logical Equivalence

Tautology: a proposition that is always true **Contradiction**: a proposition that is always false

Logically equivalent: same truth value regardless of the truth values of their individual propositions

DeMorgan's Laws:
$$\neg(p \vee q) \equiv \neg p \wedge \neg q$$
$$\neg(p \wedge q) \equiv \neg p \vee \neg q$$

Verbally,
It is not true that the patient has migraines *or* high blood pressure ≡
≡ The patient does not have migraines *and* does not have high blood pressure

It is not true that the patient has migraines *and* high blood pressure ≡
≡ The patient does not have migraines *or* does not have high blood pressure

1.5 Laws of Propositional Logic

You can use **substitution** on logically equivalent propositions.

Law Name	$\vee$ or	$\wedge$ and
Idempotent	$p \vee p \equiv p$	$p \wedge p \equiv p$
Associative	$(p \vee q) \vee r \equiv p \vee (q \vee r)$	$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$
Commutative	$p \vee q \equiv q \vee p$	$p \wedge q \equiv q \wedge p$
Distributive	$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$	$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$
Identity	$p\vee$ F $\equiv p$	$p\wedge$ T $\equiv p$
Domination	$p\vee$ T $\equiv$ T	$p\wedge$ F $\equiv$ F
Double Negation	$\neg\neg p \equiv p$	
Complement	$p \vee \neg p \equiv$ T	$p \wedge \neg p \equiv$ F
DeMorgan	$\neg(p \vee q) \equiv \neg p \wedge \neg q$	$\neg(p \wedge q) \equiv \neg p \vee \neg q$
Absorption	$p \vee (p \wedge q) \equiv p$	$p \wedge (p \vee q) \equiv p$
Conditional	$p \rightarrow q \equiv \neg p \vee q$	$p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$

1.6 Predicates and Quantifiers

Predicate: a logical statement where truth value is a function of a variable.

P(x): x is an even number. P(5): false P(2): true

Domain: the set of all possible values for a variable in a predicate.

Ex. $\mathbb{Z}^+$ is the set of all positive integers.
*If domain is not clear from context, it should be given as part of the definition of the predicate.

Quantifier: converts a predicate to a proposition.

Quantifier	Symbol	Meaning
Universal	$\forall$	"for all"
Existential	$\exists$	"there exists"

$\exists x(x + 1 < x)$ is false.

Counter Example: universally quantified statement where an element in the domain for which the predicate is false. Useful to prove a $\forall$ statement false.

1.7 Quantified Statements

Consider the two following two predicates:

$$P(x) : x \text{ is prime, } x \in \mathbb{Z}^+$$
$$O(x) : x \text{ is odd}$$

Proposition made of predicates: $\exists x(\text{P}(x) \wedge \neg\text{O}(x))$
Verbally: there exists a positive integer that is prime but is not odd.

Free Variable: a variable that is free to be any value in the domain.
Bound Variable: a variable that is bound to a quantifier.

P(x): x came to the party
S(x): x was sick

$\text{P}(x) \overset{?}{\equiv} \neg\text{S}(x)$
$\text{P}(x) \not\equiv \neg\text{S}(x)$

	P(x)	S(x)	¬S(x)
Joe	T	F	T
Theo	F	T	F
Gert	T	F	T
Sam	F	F	T

1.8 DeMorgan's law for Quantified Statements

Consider the predicate: $\mathrm{F}(x)$: "x can fly", where x is a bird. According to the DeMorgan Identity for Quantified Statements,

$$\neg\forall x \mathrm{F}(x) \equiv \exists x \neg \mathrm{F}(x)$$

"not every bird can fly ≡ "there exists a bird that cannot fly

Example using DeMorgan Identities:

$$\begin{aligned}\neg\exists x(\mathrm{P}(x) \to \neg\mathrm{Q}(x)) &\equiv \forall x\neg(\mathrm{P}(x) \to \neg\mathrm{Q}(x))\\ &\equiv \forall x(\neg\neg\mathrm{P}(x) \wedge \neg\neg\mathrm{Q}(x))\\ &\equiv \forall x(\mathrm{P}(x) \wedge \mathrm{Q}(x))\end{aligned}$$

1.9 Nested Quantifiers

A logical expression with more than one quantifier that binds different variables in the same predicate is said to have **Nested Quantifiers**.

Logic	Variable Boundedness
$\forall x \exists y$ $\mathrm{P}(x, y)$	x, y bound
$\forall x$ $\mathrm{P}(x, y)$	x bound, y free
$\exists x \exists y$ $\mathrm{T}(x, y, z)$	x, y bound, z free

Logic	Meaning
$\forall x \forall y$ $\mathrm{M}(x, y)$	"everyone sent an email to everyone"
$\forall x \exists y$ $\mathrm{M}(x, y)$	"everyone sent an email to someone"
$\exists x \forall y$ $\mathrm{M}(x, y)$	"someone sent an email to everyone"
$\exists x \exists y$ $\mathrm{M}(x, y)$	"someone sent an email to someone"

There is a two-player game analogy for how quantifiers work:

Player	Action	Goal
Existential Player $\exists$	selects value for existentially-bound variables	tries to make expression <u>true</u>
Universal Player $\forall$	selects value for universally-bound variables	tries to make expression <u>false</u>

Consider the predicate $\mathrm{L}(x, y)$: "x likes y".

$\exists x \forall y \mathrm{L}(x, y)$ means "there is a student who likes everyone in the school".

$\neg\exists x \forall y \mathrm{L}(x, y)$ means "there is no student who likes everyone in the school".

After applying DeMorgan's Laws,

$\forall x \exists y \neg \mathrm{L}(x, y)$ means "there is no student who likes everyone in the school".

1.10 More Nested Quantifiers

$\mathrm{M}(x, y)$: "x sent an email to y". Consider $\forall x \forall y$ M (x, y). It means that "email sent an email to everyone including themselves". Using $(x \neq y \to \ \mathrm{M}(x, y))$ can fix this quirk.

$\forall x \forall y(x \neq y \to \ \mathrm{M}(x, y))$ means "everyone sent an email to everyone else

1.10.1 Expressing Uniqueness in Quantified Statements

Consider $\mathrm{L}(x)$: x was late to the meeting. If someone was late to the meeting, how could you express that that someone was the only person late to the meeting? You want to express that there is someone where everyone else was not late, which can be done with

$$\exists x(\mathrm{L}(x) \wedge \forall y(x \neq y \to \neg\mathrm{L}(y)))$$

1.10.2 Moving Quantifiers in Logical Statements

Consider M(x, y): "x is married to y" and A(x): "x is an adult". One way of expressing "For every person x, if x is an adult, then there is a person y to whom x is married to" is by this statement:

$$\forall x(\ \mathrm{A}(x) \to \exists\ \mathrm{M}(x, y))$$

Since y does not appear in A(x), "$\exists y$" can be moved so that it appears just after the "$\forall$", resulting with

$$\forall x \exists y(\ \mathrm{A}(x) \to\ \mathrm{M}(x, y))$$

When doing this, keep in mind that $\forall x \exists y \not\equiv \exists y \forall x$:

$\forall x \exists y(\ \mathrm{A}(x) \to\ \mathrm{M}(x, y))$ means
for every x, if x is an adult, there exists y who is married to x.
$\exists y \forall x(\ \mathrm{A}(x) \to\ \mathrm{M}(x, y))$ means
There exists a y, such that every x who is an adult is also married to y

1.11 Logical Reasoning

Argument: a sequence of propositions, called hypothesis, followed by a final proposition, called the conclusion.

An argument is **valid** if the conclusion is true whenever the hypothesis are all true, otherwise the argument is **invalid**.

$$\text{An argument is denoted as: } \begin{array}{l} p_1 \\ p_2 \\ \vdots \\ p_n \\ \hline \therefore c \end{array} \quad \text{where} \quad \begin{array}{l} p_1, p_2, \ldots, p_n \text{ are hypothesis} \\ c \text{ is the conclusion} \end{array}$$

The argument is valid whenever the proposition $(p_1 \wedge p_1 \wedge \cdots \wedge p_n) \to c$ is a tautology. Additionally, because of the commutative law, hypothesis can be reordered without changing the argument.

$$\begin{array}{l} p \\ p \to q \\ \hline \therefore q \end{array} \equiv \begin{array}{l} p \to q \\ p \\ \hline \therefore q \end{array}$$

1.11.1 The Form of an Argument

$$\begin{array}{l} \text{It is raining today.} \\ \text{If it is raining today, then I will not ride my bike to school.} \\ \hline \therefore \text{ I will not ride my bike to school.} \end{array} \qquad \begin{array}{l} p \\ p \to q \\ \hline \therefore q \end{array}$$

$$\text{The argument is valid because its form, } \begin{array}{l} p \\ p \to q \\ \hline \therefore q \end{array} \text{ is an valid argument.}$$

$$\begin{array}{l} \text{5 is not an even number.} \\ \text{If 5 is an even number, then 7 is an even number.} \\ \hline \therefore \text{ 7 is not an even number.} \end{array} \qquad \begin{array}{l} p \\ p \to q \\ \hline \therefore q \end{array}$$

$$\text{The argument is invalid because its form, } \begin{array}{l} \neg p \\ p \to q \\ \hline \therefore \neg q \end{array} \text{ is an invalid argument.}$$

1.12 Rules of Inference with Propositions

Using truth tables to establish the validity of an argument can become tedious, especially if an argument uses a large number of variables.

$$\begin{array}{l} p \\ p \rightarrow q \\ \hline \therefore q \end{array} \quad \text{Modus Ponens} \qquad \begin{array}{l} p \\ q \\ \hline \therefore p \wedge q \end{array} \quad \text{Conjunction}$$

$$\begin{array}{l} \neg q \\ p \rightarrow q \\ \hline \therefore \neg p \end{array} \quad \text{Modus Tollens} \qquad \begin{array}{l} p \rightarrow q \\ q \rightarrow r \\ \hline \therefore p \rightarrow r \end{array} \quad \text{Hypothetical Syllogism}$$

$$\begin{array}{l} p \\ \hline \therefore p \vee q \end{array} \quad \text{Addition} \qquad \begin{array}{l} p \vee q \\ \neg p \\ \hline \therefore q \end{array} \quad \text{Disjunctive Syllogism}$$

$$\begin{array}{l} p \wedge q \\ \hline \therefore p \end{array} \quad \text{Simplification} \qquad \begin{array}{l} p \rightarrow q \\ q \rightarrow r \\ \hline \therefore q \vee r \end{array} \quad \text{Resolution}$$

Example expressed in English:

$$\begin{array}{l} \text{If it is raining or windy or both, the game will be cancelled.} \\ \text{The game will not be cancelled} \\ \hline \therefore \text{ It is not windy.} \end{array} \qquad \begin{array}{l} (r \vee w) \rightarrow c \\ \neg c \\ \hline \therefore \neg w \end{array}$$

Steps to Solve:

$(r \vee w) \rightarrow c$	Hypothesis	(1)
$\neg c$	Hypothesis	(2)
$\neg(r \vee w)$	Modus Tollens: 1, 2	(3)
$\neg r \wedge \neg w$	DeMorgan's Law: 3	(4)
$\neg w \wedge \neg r$	Commutative Law: 4	(5)
$\neg w$	Simplification: 5	(6)

1.13 Rules of Inference with Quantifiers

In order to apply the rules of quantified expressions, such as $\forall x \neg(\text{ P}(x) \wedge \text{ Q}(x))$, we need to remove the quantifier by plugging in a value from the domain to replace the variable x.

For example:

$$\begin{array}{l} \text{Every employee who received a large bonus works hard.} \\ \text{Linda is an employee at the company.} \\ \text{Linda received a large bonus.} \\ \hline \therefore \text{ Some employee works hard.} \end{array} \qquad \begin{array}{l} \forall x(\text{B}(x) \rightarrow \text{ H}(x)) \\ \text{Linda} \in x \\ \text{B(Linda)} \\ \hline \therefore \exists x \text{ H}(x) \end{array}$$

Arbitrary Element: has no special properties other than those shared by all elements of the domain.

Particular Element: may have special properties that are not shared by all the elements of the domain. For example, if the domain is the set of all integers, $\mathbb{Z}$, a particular element is 3, because it is odd, which is not true for all integers.

Rules of Inference for Quantified Statements

$$\begin{array}{l} c \text{ is an element} \\ \forall x \ \mathrm{P}\ (x) \\ \hline \therefore \mathrm{P}(c) \end{array} \quad \text{Universal Instantiation} \qquad \begin{array}{l} \exists x \ \mathrm{P}(x) \\ \hline \therefore c \text{ is particular} \wedge \mathrm{P}\ (c) \end{array} \quad \text{Existential Instantiation*}$$

$$\begin{array}{l} c \text{ is arbitrary} \\ \mathrm{P}\ (c) \\ \hline \therefore \forall\ \mathrm{P}(x) \end{array} \quad \text{Universal Generalization} \qquad \begin{array}{l} c \text{ is an element} \\ \mathrm{P}(c) \\ \hline \therefore \exists x \ \mathrm{P}(x) \end{array} \quad \text{Existential Generalization}$$

*Each use of Existential Instantiation must define a new element with its own symbol or name.

1.13.1 Example of using the Laws of Inference for Quantified Statements

Consider the following argument:
$$\begin{array}{l} \forall x(\mathrm{P}(x) \vee \ \mathrm{Q}(x)) \\ 3 \text{ is an integer} \\ \neg\ \mathrm{P}(3) \\ \hline \therefore\ \mathrm{Q}(3) \end{array}$$

Steps to Solve:

$\forall x(\mathrm{P}(x) \vee \ \mathrm{Q}(x))$	Hypothesis	(1)
3 is an integer	Hypothesis	(2)
$(\mathrm{P}(3) \vee \mathrm{Q}(3))$	Universal Instantiation: 1, 2	(3)
$\neg\ \mathrm{P}(3)$	Hypothesis	(4)
$\mathrm{Q}(3)$	Disjunctive Syllogism: 3, 4	(5)

1.13.2 Showing an Argument with Quantified Statements is Invalid

Consider the following argument:
$$\begin{array}{l} \exists x \mathrm{P}(x) \\ \exists x \mathrm{Q}(x) \\ \hline \therefore \exists x(\mathrm{P}(x) \wedge \ \mathrm{Q}(x)) \end{array}$$

Using a supposed domain $\{c, d\}$, with truth values of

	P	Q
c	T	F
d	F	T

, the argument is invalid.

2 Proofs

2.1 Mathematical Definitions

2.2 Introduction to Proofs

2.3 Writing Proofs: Best Practices

2.4 Writing Direct Proofs

2.5 Proof by Contrapositive

2.6 Proof by Contradiction

2.7 Proof by Cases

3 Sets

3.1 Sets and Subsets

A **set** is a collection of objects. Objects in a set are called **elements**. Order does $\underline{\text{not}}$ matter, and there are $\underline{\text{no}}$ duplicates.

Roster notation:

$$\begin{aligned} A &= \{2,4,6,10\} \\ B &= \{4,6,10,2\} \\ A &= B \end{aligned}$$

To show membership, use the $\in$ symbol. For example, $2 \in A$, while $7 \notin A$. The empty set, which contains nothing, typically uses the $\emptyset$ symbol, or $\{\}$. Sets can be finite, or infinite. **Cardinality** of a set is the number of elements in a set. For example, the cardinality of A is 4.

$$|A| = 4$$

Cardinality can be infinite. Consider the set of all the integers, $\mathbb{Z}$. $|\mathbb{Z}| = \infty$

$$\begin{aligned} \mathbb{N} :& \text{ set of natural numbers} \\ =& \{0,1,2,3,\ldots\} \end{aligned}$$

$$\begin{aligned} \mathbb{Z} :& \text{ set of integers} \\ =& \{\ldots,-2,-1,0,1,2,\ldots\} \end{aligned}$$

$$\begin{aligned} \mathbb{Q} :& \text{ set of rational numbers} \\ =& \{x | x = \frac{a}{b} \text{ where } a, b \in \mathbb{Z}, b \neq 0\} \end{aligned}$$

$$\begin{aligned} \mathbb{R} :& \text{ set of real numbers} \\ =& \{x | x \text{ has a decimal representation}\} \end{aligned}$$

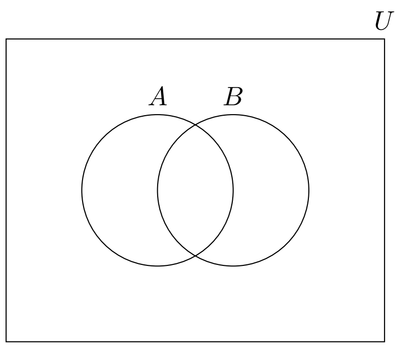
The subset operator is $\subseteq$

$$\begin{aligned} &A \subseteq B \text{ if } \forall x (x \in A \rightarrow x \in B) \\ &A \subseteq A \text{ is true for } \underline{\text{any}} \text{ set} \\ &\emptyset \subseteq A \text{ is true for } \underline{\text{any}} \text{ set} \end{aligned}$$

Sometimes it is easier to define a set by defining properties that all the elements have. That is easy to do in **set builder notation**.

$$\begin{aligned} A &= \{x \in S : P(x)\}, \text{ where } S \text{ is another set} \\ C &= \{x \in \mathbb{Z} : 0 < x < 100 \text{ and } x \text{ is prime}\}. \\ D &= \{x \in \mathbb{R} : |x| < 1\} \end{aligned}$$

The **Universal Set**, usually called 'U', is a set that contains all elements mentioned in a particular context. For example, a discussion about certain types of real numbers, it would be understood that any element in the discussion is a real number. Sets are often represented pictorially with **Venn Diagrams**.



If $A \subseteq B$ and there is an element of B that is not an element of A, meaning $A \neq B$, then A is a **proper subset** of B, denoted as $A \subset B$. An important fact is that

$$\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R}$$

3.2 Sets of sets

3.3 Union and Intersection

3.4 More set operations

3.5 Set identities

3.6 Cartesian products

3.7 Partitions

4 Functions

4.1 Definition of functions

4.2 Floor and Ceiling functions

4.3 Properties of functions

4.4 The inverse of a function

4.5 Composition of functions

4.6 Logarithms and exponents

5 Boolean Algebra

5.1 An introduction to Boolean Algebra

5.2 Boolean functions

5.3 Disjunctive and conjunctive normal form

5.4 Functional completeness

5.5 Boolean satisfiability

5.6 Gates and circuits

6 Relation and Digraphs

6.1 Introduction to binary relations

6.2 Properties of binary relations

6.3 Directed graphs, paths, and cycles

6.4 Composition of relations

6.5 Graph powers and the transitive closure

6.6 Matrix multiplication and graph powers

6.7 Partial orders

6.8 Strict orders and directed acyclic graphs

6.9 Equivalence relations

6.10 N-ary relations and relational databases

7 Computation

7.1 An introduction to algorithms

7.2 Asymptotic growth of functions

7.3 Analysis of algorithms

7.4 Finite state machines

7.5 Turing machines

7.6 Decision problems and languages

8 Induction and Recursion

8.1 Sequences

8.2 Recurrence relations

8.3 Summations

8.4 Mathematical induction

8.5 More inductive proofs

8.6 Strong induction and well-ordering

8.7 Loop invariants

8.8 Recursive definitions

8.9 Structural induction

8.10 Recursive algorithms

8.11 Induction and recursive algorithms

8.12 Analyzing the time complexity of recursive algorithms

8.13 Divide-and-conquer algorithms: Introduction and mergesort

8.14 Divide-and-conquer algorithms: Binary Search

8.15 Solving linear homogeneous recurrence relations

8.16 Solving linear non-homogeneous recurrence relations

8.17 Divide-and-conquer recurrence relations

9 Integer Properties

9.1 The Division Algorithm

9.2 Modular arithmetic

9.3 Prime factorizations

9.4 Factoring and primality testing

9.5 Greatest common factor divisor and Euclid's algorithm

9.6 Number representation

9.7 Fast exponentiation

9.8 Introduction to cryptography

9.9 The RSA cryptosystem

10 Introduction to Counting

10.1 Sum and Product Rules

10.2 The Bijection Rules

10.3 The generalized product rule

10.4 Counting permutations

10.5 Counting subsets

10.6 Subset and permutation examples

10.7 Counting by complement

10.8 Permutations with repetitions

10.9 Counting multisets

10.10 Assignment problems: Balls in bins

10.11 Inclusion-exclusion principle

11 Advanced Counting

11.1 Generating permutations

11.2 Binomial coefficients and combinatorial identities

11.3 The pigeonhole principle

11.4 Generating functions

12 Discrete Probability

12.1 Probability of an event

12.2 Unions and complements of events

12.3 Conditional probability and independence

12.4 Bayes' Theorem

12.5 Random variables

12.6 Expectation of random variables

12.7 Linearity of expectations

12.8 Bernoulli trials and the binomial distribution

13 Graphs

13.1 Introduction to Graphs

13.2 Graph representations

13.3 Graph isomorphism

13.4 Walks, trails, circuits, paths, and cycles

13.5 Graph connectivity

13.6 Euler circuits and trails

13.7 Hamiltonian cycles and paths

13.8 Planar coloring

13.9 Graph coloring

14 Trees

14.1 Introduction to trees

14.2 Tree application examples

14.3 Properties of trees

14.4 Tree traversals

14.5 Spanning trees and graph traversals

14.6 Minimum spanning trees