A Brief Guide To Discrete Structures

Zach Harel

November 12, 2025

Contents

Chapter 1

Logic

1.1 Propositional Logic

A **proposition** is a statement that is either true or false.

1.1.1 Basic Logical Operators

Everything in propositional logic can be created using only NOT, AND, and OR.

$\neg$: Negation (NOT)

The negation of p ($\neg p$) is true when p is false, and false when p is true.

$\wedge$: Conjunction (AND)

$p \wedge q$ is true only when both p and q are true.

$\vee$: Disjunction (OR)

$p \vee q$ is true when at least one of p or q is true.

1.1.2 More Logical Operators

These operators are commonly used in both mathematics and computer science due to their utility. However, they are simply combinations of the three basic operators.

$\implies$: Implication (IF-THEN)

$p \implies q$ is false only when p is true and q is false.

- p is the **hypothesis** (or antecedent)
- q is the **conclusion** (or consequent)

$$p \implies q \equiv \neg p \vee q \tag{1.1}$$

$\oplus$: Exclusive Disjunction (XOR)

$p \oplus q$ is true when either p is true or q is true, but not both.

$$p \oplus q \equiv (p \vee q) \wedge \neg(p \wedge q) \tag{1.2}$$

1.1.3 Logical Equivalence

Two propositions are **logically equivalent** ($\equiv$) if they have the same truth value for all possible truth assignments.
There are many laws of logical equivalence; you can Google them, but here are some commonly used ones.

De Morgan's Laws

$$\neg(p \wedge q) \equiv \neg p \vee \neg q \tag{1.3}$$
$$\neg(p \vee q) \equiv \neg p \wedge \neg q \tag{1.4}$$

Double Negation

$$\neg(\neg p) \equiv p \tag{1.5}$$

Contrapositive

$$(p \implies q) \equiv (\neg q \implies \neg p) \tag{1.6}$$

1.2 First-Order Logic

First-order logic extends propositional logic with **predicates** and **quantifiers**.

1.2.1 Predicates

A **predicate** is a statement containing variables that becomes a proposition when variables are assigned values.
Example: $P(x)$: "x is even" becomes a proposition when x is specified.

1.2.2 Quantifiers

$\forall$: Universal Quantifier (FOR ALL)

$\forall x\, P(x)$ means $P(x)$ is true for every value of x in the domain.

$\exists$: Existential Quantifier (THERE EXISTS)

$\exists x\, P(x)$ means there is at least one value of x in the domain for which $P(x)$ is true.

1.2.3 Negating Quantifiers

$$\neg(\forall x\, P(x)) \equiv \exists x\, \neg P(x) \tag{1.7}$$
$$\neg(\exists x\, P(x)) \equiv \forall x\, \neg P(x) \tag{1.8}$$

1.2.4 Multiple Quantifiers

Order matters! $\forall x\, \exists y\, P(x, y)$ is different from $\exists y\, \forall x\, P(x, y)$.
Example: Let $P(x, y)$: "$x < y$"

- $\forall x\, \exists y\, P(x, y)$: For every number, there exists a larger number (TRUE)
- $\exists y\, \forall x\, P(x, y)$: There exists a number larger than all numbers (FALSE)

Chapter 2

Representation of Numbers

2.1 Numeric Bases

A number system with base b uses digits 0 through $b-1$ to represent values. The rightmost digit represents b^0, the next represents b^1, and so on.

2.1.1 Common Bases

- **Binary** (base 2): Uses digits 0, 1
- **Octal** (base 8): Uses digits 0-7
- **Decimal** (base 10): Uses digits 0-9
- **Hexadecimal** (base 16): Uses digits 0-9, A-F (where A=10, B=11, ..., F=15)

2.1.2 Converting From Base b to Decimal

To convert a number in base b to decimal, multiply each digit by its positional value and sum.

$$(d_n d_{n-1} \dots d_1 d_0)_b = d_n \cdot b^n + d_{n-1} \cdot b^{n-1} + \dots + d_1 \cdot b^1 + d_0 \cdot b^0 \tag{2.1}$$

Example: $(1011)_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 8 + 0 + 2 + 1 = (11)_{10}$

2.1.3 Converting From Decimal to Base b

Repeatedly divide by b and record remainders from bottom to top.
Example: Convert $(13)_{10}$ to binary:

$$\begin{aligned} 13 \div 2 &= 6 \text{ remainder } 1 \\ 6 \div 2 &= 3 \text{ remainder } 0 \\ 3 \div 2 &= 1 \text{ remainder } 1 \\ 1 \div 2 &= 0 \text{ remainder } 1 \end{aligned}$$

Reading remainders from bottom to top: $(13)_{10} = (1101)_2$

2.1.4 Converting Between Non-Decimal Bases

Convert through decimal as an intermediate step, or use direct conversion for related bases (e.g., binary $\leftrightarrow$ hexadecimal).
Binary to Hexadecimal: Group binary digits in sets of 4 (starting from right), convert each group to its hex equivalent.
Example: $(11010110)_2 = (1101)(0110)_2 = (D6)_{16}$

2.2 Signed and Unsigned Numbers

2.2.1 Unsigned Representation

All bits represent magnitude. An n-bit unsigned number can represent values from 0 to $2^n - 1$.
Example: With 8 bits, unsigned range is 0 to 255.

2.2.2 Signed Representation

Multiple methods exist to represent signed integers:

Sign-Magnitude

The leftmost bit is the sign bit (0 for positive, 1 for negative), remaining bits represent magnitude.
Problems: Two representations of zero (+0 and −0), arithmetic is complex.

One's Complement

Negative numbers are formed by flipping all bits of the positive representation.
Problems: Still has two zeros, arithmetic requires end-around carry.

2.3 Two's Complement

Two's complement is the standard representation for signed integers in modern computers.

2.3.1 Definition

For an n-bit number:

- If the leftmost bit is 0, the number is positive (same as unsigned)
- If the leftmost bit is 1, the number is negative

An n-bit two's complement number can represent values from -2^{n-1} to $2^{n-1} - 1$.

2.3.2 Converting to Two's Complement

To represent $-x$ in two's complement:

1. Write the binary representation of x
2. Flip all bits (one's complement)
3. Add 1

Example: Find −5 in 8-bit two's complement:

$$\begin{aligned} 5 &= (00000101)_2 \\ \text{Flip bits:} \quad &= (11111010)_2 \\ \text{Add 1:} \quad &= (11111011)_2 \end{aligned}$$

Therefore, $-5 = (11111011)_2$ in 8-bit two's complement.

2.3.3 Converting From Two's Complement

If the leftmost bit is 1 (negative number), apply the same process: flip all bits and add 1.
Example: What is $(11111011)_2$ in decimal?

$$\begin{aligned} \text{Flip bits:} &= (00000100)_2 \\ \text{Add 1:} &= (00000101)_2 = 5 \\ \text{Since original was negative:} &= -5 \end{aligned}$$

2.3.4 Properties of Two's Complement

- Only one representation of zero: $(00...0)_2$
- Range for n bits: -2^{n-1} to $2^{n-1} - 1$
- The negative of a number equals its two's complement
- The leftmost bit has weight -2^{n-1}

2.4 Arithmetic in Two's Complement

2.4.1 Addition

Add numbers normally using binary addition. Discard any carry out of the leftmost bit.
Example: $5 + (-3) = 2$ in 8-bit two's complement:

$$\begin{array}{r} 00000101 \text{ (5)} \\ +\,11111101 \text{ (-3)} \\ \hline 1\,00000010 \text{ (discard carry, result = 2)} \end{array}$$

2.4.2 Subtraction

To compute $a - b$, add a and the two's complement of b: $a + (-b)$.
Example: $7 - 5 = 7 + (-5)$:

$$\begin{array}{r} 00000111 \text{ (7)} \\ +\,11111011 \text{ (-5)} \\ \hline 1\,00000010 \text{ (result = 2)} \end{array}$$

2.4.3 Overflow Detection

Overflow occurs when the result exceeds the representable range.
Overflow conditions:

- Adding two positive numbers yields a negative result
- Adding two negative numbers yields a positive result
- Equivalent: carry into the sign bit $\neq$ carry out of the sign bit

Example: $127 + 1$ in 8-bit (overflow):

$$\begin{array}{r} 01111111 \text{ (127)} \\ +\,00000001 \text{ (1)} \\ \hline 10000000 \text{ (-128, overflow!)} \end{array}$$

2.4.4 Sign Extension

To represent an n-bit two's complement number with more bits, copy the sign bit into all new leftmost positions.
Example: Extend (-5) from 8-bit to 16-bit:

8-bit: 11111011
16-bit: 1111111111111011

Chapter 3

Sets

3.1 What is a Set?

A set is simply an unordered collection of unique items.

3.2 Default Sets

3.2.1 $\mathbb{U}$: The Universal Set

Contains every element in our domain (must be defined for each problem/set of problems).

3.2.2 $\varnothing$: The Empty Set

{}; Has no elements.

3.3 Set Operations

3.3.1 $=$: Equality

Two sets are equal if all of their elements are the same.

$$(A = B) \iff (\forall x(x \in A \iff x \in B)) \tag{3.1}$$

3.3.2 $|S|$: Cardinality

The cardinality of set S ($|S|$) is the number of elements in set S.

$$|\{1, 2, 3\}| = 3 \tag{3.2}$$

3.3.3 $\overline{S}$: Complement

The complement of a set S ($\overline{S}$) is the set of every element (of the universal set) that is not in S.
For example, if $\mathbb{U} = \{1, 2, 3, 4, 5\}$ and $S = \{1, 4\}$, $\overline{S} = \{2, 3, 5\}$.

$$|\overline{S}| = |\mathbb{U}| - |S| \tag{3.3}$$

3.3.4 $\cap$: Intersection

The intersection of sets A and B ($A \cap B$) contains every element in both sets.

$$A \cap B = \{x | x \in A \wedge x \in B\} \tag{3.4}$$

3.3.5 $\cup$: Union

The union of sets A and B ($A \cup B$) contains every element in either set (with no repeats).

$$A \cup B = \{x | x \in A \vee x \in B\} \tag{3.5}$$
$$|A \cup B| = |A| + |B| + |A \cap B| \tag{3.6}$$

3.3.6 $\subseteq$: Subset

If A is a subset of B ($A \subseteq B$), all elements of A are also elements of B.

$$A \subseteq B \iff \forall x (x \in A) \implies (x \in B) \tag{3.7}$$

Any set is a subset of itself.

$$A \subseteq A \tag{3.8}$$

The subset definition can also be used to define set equality.

$$A = B \iff (A \subseteq B) \wedge (B \subseteq A) \tag{3.9}$$

If A is a *proper* (also called strict) subset of B ($A \subset B$), all elements of A are in B, but A is not B.

$$A \subset B \iff (A \subseteq B) \wedge (A \neq B) \tag{3.10}$$

If A is a subset of B, B is a superset of A; if A is a proper subset of B, B is a proper superset of A.

$$A \subseteq B \iff B \supseteq A \tag{3.11}$$
$$A \subset B \iff B \supset A \tag{3.12}$$

3.3.7 $-$: Difference

The difference of sets A and B (A - B) is every element of A that is not in B.

$$A - B = A \cap \overline{B} \tag{3.13}$$
$$|A - B| = |A \cap \overline{B}| \tag{3.14}$$

Set difference can also be used to define a set's absolute complement (the previously defined complement):

$$\overline{S} = \mathbb{U} - S \tag{3.15}$$

The difference of sets A and B ($A - B$) is sometimes called the relative complement of B with respect to A ($B \backslash A$). The absolute complement of a set S ($\overline{S}$)is simply its relative complement with respect to the universal set ($\mathbb{U} \backslash S$).

$$A - B = A \backslash B \tag{3.16}$$

The *symmetric* difference of sets A and B ($A \Delta B$) is every element of one set that is not in the other set.

$$A \Delta B = (A - B) \cup (B - A) \tag{3.17}$$

3.4 Set Functions

3.4.1 Powerset

- input: set
- output: set of sets

$$\mathcal{P}(S) = \{A | A \subseteq S\} \tag{3.18}$$

3.4.2 Cartesian Product

- input: two sets
- output: set of ordered pairs

$$A \times B = \{(a, b) | a \in A \wedge b \in B\} \tag{3.19}$$

If $A \cap B = \varnothing$, A and B are **disjoint**.

Chapter 4

Counting

	No Repetition	Yes Repetition
Order Matters	Permutation P_k^n	n^k
Order Does Not Matter	Combination C_k^n	Stars & Bars

Table 4.1: Which Formula To Use

4.1 Counting We've Seen

$$\begin{gathered}\text{let } A = \{1,2,3\} \text{ and } B = \{3,4\} \\ \therefore \\ A \times B = \{(1,3),(1,4),(2,3),(2,4),(3,3),(3,4)\}\end{gathered}$$

$$|A \times B| = |A| \cdot |B| \tag{4.1}$$

$$|\mathcal{P}(S)| = 2^{|S|} \tag{4.2}$$

4.2 Product Rule: Order Matters

4.2.1 Two Separate Tasks

- one task in n ways
- one task in m ways

$$\text{There are } n \cdot m \text{ ways to do task 1 } \textbf{and} \text{ task 2.} \tag{4.3}$$

4.2.2 One Task Multiple Ways

When repetition is okay, we have k tasks and n choices per task; there are n^k ways to do it.
When repetition is not okay, we have k tasks and n choices for the first task, $n-1$ for the second task, etc. For this, we use permutations: P_k^n.

$$\mathrm{P}_k^n = k \text{ permutations of } n \text{ objects is an ordering of } k \text{ of the objects.} \tag{4.4}$$

$$\mathrm{P}_k^n = \frac{n!}{(n-k)!};\ \mathrm{P}_n^n = n! \tag{4.5}$$

4.3 Sum Rule: Order Does Not Matter

4.3.1 Two Tasks

- one task in n ways
- one task in m ways

$$\text{There are } n+m \text{ ways to do task 1 } \textbf{or} \text{ task 2.} \tag{4.6}$$

4.3.2 One Task Multiple Ways

When repetition is okay, see the Stars and Bars subsection!
When repetition is not okay, we have k tasks and n choices for the first task, $n-1$ for the second task, etc. For this, we use combinations: C_k^n.

$$\mathrm{C}_k^n = k \text{ combinations of } n \text{ objects is a set of } k \text{ of the objects.} \tag{4.7}$$

$$\mathrm{C}_k^n = \frac{n!}{k!(n-k)!};\ \mathrm{C}_n^n = 1 \tag{4.8}$$

4.3.3 Stars and Bars

1. Order doesn't matter
2. Repetition is allowed

How many ways are there to put n indistinguishable balls into n distinguishable bins?
This is secretly a combination problem!

$$n_{\text{combination}} = n_{\text{stars and bars}} + k_{\text{stars and bars}} + 1 \tag{4.9}$$

$$k_{\text{combination}} = k_{\text{stars and bars}} - 1 \tag{4.10}$$

This means that the answer to our initial question is simply C_{k-1}^{n+k+1}

4.4 The Pigeonhole Principle

The bridge between Counting and Probability! We are putting objects in boxes.
This principle starts with the idea that you have a bunch of pigeons, and a bunch of pigeonholes, and every pigeon must go into a pigeonhole. If there are more pigeons than pigeonholes, then you are guaranteed that at least one pigeonhole contains at least 2 pigeons. This is the closest I could get to drawing pigeons; there are 6 pigeonholes in the drawings below and 7 pigeons.
If there are 200 people in a room, it is guaranteed that at least 17 of them share a birth month. It is not guaranteed (it is likely) that even one person was born in October (or any other month, for that matter).
In more general terms, the Pigeonhole Principle states that, if you have n objects and $k < n$ boxes, there must be at least one box with $\lceil \frac{n}{k} \rceil$ [1] objects in it.

$$\text{Given } n \text{ objects and } k \text{ boxes, at least one box has } \left\lceil \frac{n}{k} \right\rceil \text{ objects in it.} \tag{4.11}$$

[1] $\lceil x \rceil$ aka ceil(x) is the smallest integer that is greater than x. Eg $\lceil 2.5 \rceil = \lceil 2.9 \rceil = \lceil 2.00001 \rceil = 3$.

Chapter 5

Probability

5.1 Terms

Experiment

An infinitely repeatable procedure with a well-defined set of outcomes.

Sample Space S

All possible outcomes of an experiment.

$$S = \{s_1, s_2, s_3, ...s_n\} \tag{5.1}$$

Event Space E

Subset of outcomes we care about.

$$E \subseteq S \tag{5.2}$$

Probability of Event E

How likely event E is to occur.

$$\Pr(E) = \frac{|E|}{|S|} \tag{5.3}$$

$$\Pr(\neg E) = \frac{|\overline{E}|}{|S|} = 1 - \Pr(E) \tag{5.4}$$

The RHS of equation 5.4 works because the sample space S is the universal set of events. Note that despite treating E like a set, the logical not ($\neg$) symbol is used to mark an event *not* occurring.

Expected Value x

A numeric value associated with the outcome of the experiment. Used because we can't take the average of 'pink sock' or 'queen of hearts'.

5.2 Probability of Multiple Events

A generalization of the product rule and sum rule.
Given events E and F:

$$\Pr(E \cup F) = \Pr(E) + \Pr(F) + \Pr(E \cap F) \tag{5.5}$$

5.2.1 Independent Events

Two events are independent if the outcome of one doesn't affect the outcome of the other.

$$\Pr(E \cap F) = \Pr(E) \cdot \Pr(F) \iff E \text{ and } F \text{ are independent} \tag{5.6}$$

In addition to determining the probability of two events that we *know* are independent, we can also use that formula to determine *if* two events are independent.

5.2.2 Conditional Probability

The likelihood of an event E occurring, given that another event F has already happened. The probability of E given F is written as $\Pr(E|F)$.

$$\Pr(E|F) = \frac{\Pr(E \cap F)}{\Pr(F)} \tag{5.7}$$

Note that when calculating $\Pr(E \cap F)$, E and F are *not* independent; formula 5.6 does not apply. However, we can arrange that equation to find that

$$\Pr(E \cap F) = \Pr(E|F) \cdot \Pr(F) \tag{5.8}$$

5.2.3 Bayes' Theorem

Bayes' theorem states

$$\Pr(E|F) = \frac{\Pr(F|E) \cdot \Pr(E)}{\Pr(F)} \tag{5.9}$$

5.2.4 General Insights

Given that $\Pr(F) = \Pr(F \cap E) + \Pr(F \cap \neg E)$ and what was discovered in equation 5.8, we can find that

$$\Pr(F) = \Pr(F|E) \cdot \Pr(E) + \Pr(F|\neg E) \cdot \Pr(\neg E) \tag{5.10}$$

5.3 Expectations

5.3.1 Expected Values

After running an experiment over and over...

On average, what happens?

$$E[x] = \sum_{s_i \in S} \Pr(s_i) \cdot x_i \tag{5.11}$$

It is essentially a weighted average of each outcome times the value of that outcome.

5.3.2 Linearity of Expectations

What if we wanted to know the number of heads if we flip a coin 100 times?

$$E(x_1 + x_2 + \cdots + x_n) = E[x_1] + E[x_2] + \cdots + E[x_n] \tag{5.12}$$

Chapter 6

Sequences and Series

A sequence is a discrete structure used to discover and characterize patterns (such as earthquakes, comets, and the stock market).

A sequence is an ordered list of potentially an infinite number of numbers.

6.1 Definition

A sequence is defined like $\{a_n\} = a_1, a_2, a_3, \cdots, a_n$. If the sequence is infinite, there is no nth term.

An infinite sequence is also defined as a mapping from $\mathbb{Z}^+$ to $\{a_n\}$.

6.1.1 Notation

- a_n: the whole sequence or the final term
- a_k: any one arbitrary term

6.1.2 Forms

Recursive Form

A sequence can be defined recursively, using the previous term.

An arithmetic sequence can be defined recursively like this:

$$a_k = a_{k-1} + \beta \tag{6.1}$$

A geometric sequence can be defined recursively like this:

$$a_k = \beta \cdot a_{k-1} \tag{6.2}$$

Closed Form

A closed form sequence does not depend on any other term.

An arithmetic sequence can be defined explicitly like this:

$$a_k = \beta \cdot k \tag{6.3}$$

A geometric sequence can be defined recursively like this:

$$a_k = \beta^k \tag{6.4}$$

Of course, other types of sequences exist, such as:

$$a_n \forall n \in \mathbb{Z}^+ a_k = \frac{k}{k+1}$$

With this sequence, $a_1 = \frac{1}{2}$, $a_2 = \frac{2}{3}$, and $a_{512} = \frac{512}{513}$.

And

$$b_n \forall n \in \mathbb{Z}^+ b_k = (-1)^k$$

With this sequence, $b_1 = -1$, $b_2 = 1$, and $b_{124} = 1$.

6.2 Specific Series

6.2.1 Arithmetic Series

For any arithmetic series $a_k = a + (k-1)d$, the sum of the first n terms is

$$\sum_{k=1}^{n} a_k = \frac{n(a_1 \cdot a_n)}{2} \tag{6.5}$$

6.2.2 Geometric Series

For any geometric series $a_k = a \cdot r^{k-1}$, the sum of the first n terms is

$$\sum_{k=1}^{n} a_k = \frac{a_1 \cdot r^n - a_1}{r-1} \tag{6.6}$$

6.3 Summation Rules

Constant Summation

$$\sum_{i=1}^{n} k = nk \tag{6.7}$$

where $k \in \mathbb{R}$

Constant Multiplication

$$\sum_{i=1}^{n} a_i = k \sum_{i=1}^{n} a_i \tag{6.8}$$

where $k \in \mathbb{R}$ and a_i is the ith term of a sequence $\{a_n\}$.

Summation Bounds

$$\sum_{i=a}^{n} s_i = \sum_{i=1}^{n} s_i + \sum_{i=1}^{a-1} s_i \tag{6.9}$$

where $a \in \mathbb{N}$ and s_i is the ith term of a sequence $\{s_n\}$.

Summation Addition

$$\sum_{i=1}^{n} a_i + b_i = \sum_{i=1}^{n} a_i + \sum_{i=1}^{n} b_i \tag{6.10}$$

where a_i and b_i are the ith term of sequences $\{a_n\}$ and $\{b_n\}$ respectively.

Chapter 7

Mathematical Induction

Proof by induction is a technique used in mathematics and computer science. Given some predicate $P(x)$, a proof by induction is a shortcut to prove $\forall n \in \mathbb{Z}^+$, $P(n)$ by proving that $P(k) \implies P(k+1)$.

7.1 Structure

A proof by induction has some major components:

7.1.1 Predicate and Logic Statement

The first part is the predicate, which is usually an equation or statement that may or may not be true for any given value of its inputs. Some examples include:

- $P(x) : \sum_{i=1}^{x} i = \frac{i(i+1)}{2}$
- $P(x) : 6^x + 4$ is a multiple of 5

We then create a logic statement so that we can actually prove it:

$$\forall n \in \mathbb{Z}^+, P(n) \tag{7.1}$$

7.1.2 Inductive Hypothesis and Implication

To prove that the logic statement is true, we prove the implication

$$P(k) \implies P(k+1) \tag{7.2}$$

and assume $P(k)$ is true. That assumption is called the inductive hypothesis.

7.1.3 Base Case

The base case is a proof for the smallest value of x that we want to prove. The inductive cases will follow it.

7.1.4 Inductive Step

Now that we assume $P(k)$ is true and we've proven $P(x)$ for the smallest value of x, we can use it to prove $P(k+1)$.

7.2 Example: Integer Summations

Proof. We prove by mathematical induction that $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$ for all $n \in \mathbb{Z}^+$.

Base case: For $n = 1$:

$$\sum_{i=1}^{1} i = 1 = \frac{(1)(1+1)}{2} = \frac{2}{2} \tag{7.3}$$

The base case holds.

Inductive step: Assume the inductive hypothesis that for some $k \in \mathbb{Z}^+$:

$$\sum_{i=1}^{k} i = \frac{k(k+1)}{2} \tag{7.4}$$

We must show:

$$\sum_{i=1}^{k+1} i = \frac{(k+1)(k+2)}{2} \tag{7.5}$$

Starting with the left-hand side:

$$\begin{aligned}
\sum_{i=1}^{k+1} i &= \sum_{i=1}^{k} i + (k+1) \\
&= \frac{k(k+1)}{2} + (k+1) \quad \text{(by inductive hypothesis)} \\
&= \frac{k(k+1)}{2} + \frac{2(k+1)}{2} \\
&= \frac{k(k+1) + 2(k+1)}{2} \\
&= \frac{(k+1)(k+2)}{2}
\end{aligned}$$

By mathematical induction, the theorem holds for all positive integers n. □