MATHEMATICS FOR COMPUTER SCIENCE

Hao Zhang
zhangh0214@gmail.com

0 Preface

This note explains how to use mathematical models and methods to analyze problems that arise in computer science.

Acknowledges

[1]. Thomas Cormen, Charles Leiserson, Ronald Rivest, and Clifford Stein. Introduction to Algorithms. MIT press Cambridege. 2009.

[2]. Eric Lehman, Thomson Leighton, and Albert Meyer. Mathematics for Computer Science (2010 version). MIT. 2010.

[3]. Eric Lehman, Thomson Leighton, and Albert Meyer. Mathematics for Computer Science (2017 version). MIT. 2017.

Contents

List of Figures

List of Tables

List of Algorithms

I PROOFS

1 Propositions

To get around the ambiguity of English, mathematicians have devised a special language for talking about logical relationships.

1.1 Propositional Formulas

1.1.1 Basic Concepts

DEFINITION 1.1 **Proposition** A statement (communication) that is either true or false.

Proposition excludes statements whose truth varies with circumstance.

DEFINITION 1.2 **Propositional Variables/Boolean Variables** Variables, like propositions, can take on only the values T (true) and F (false).

DEFINITION 1.3 **Truth Assignments/Environment** A truth assignments/environment assigns a value T (true) or F (false) to each propositional variable.

DEFINITION 1.4 **Truth Table** A truth table indicates the true/false value of a proposition for each possible environment of the variables.

1.1.2 Propositions from Propositions

DEFINITION 1.5 **And** $P \wedge Q$

DEFINITION 1.6 **Or** $P \vee Q$

DEFINITION 1.7 **Not** $\neg P$

DEFINITION 1.8 **Nor** $\neg(P \vee Q)$ Negation of $P \vee Q$.

$$\neg(P \vee Q) := \neg P \wedge \neg Q\,. \tag{1.1}$$

DEFINITION 1.9 **Exclusive-or/XOR** $P \oplus Q$ $P \oplus Q$ is true exactly when exactly one of P and Q is true.

$$P \oplus Q := (P \wedge \neg Q) \vee (\neg P \wedge Q) \equiv (P \Rightarrow \neg Q) \wedge (\neg P \Rightarrow Q)\,. \tag{1.2}$$

DEFINITION 1.10 **Implies** $P \Rightarrow Q$ $P \Rightarrow Q$ is true exactly when P is false or Q is true.

$$P \Rightarrow Q := \neg P \vee Q\,. \tag{1.3}$$

False hypotheses comes from the fact that, when a system obeys the specification which is consisted of a series of rules, the and of these rules are always true.

$$\wedge_{i=1}^{n} (C_i \Rightarrow A_i)\,. \tag{1.4}$$

DEFINITION 1.11 **If and Only If/IFF** $P \Leftrightarrow Q$ $P \Leftrightarrow Q$ asserts that P and Q have the same truth value. Either both are true or both are false.

$$P \Leftrightarrow Q := (P \wedge Q) \vee (\neg P \wedge \neg Q) \equiv (P \Rightarrow Q) \wedge (Q \Rightarrow P) \tag{1.5}$$

1.1.3 Propositions in Normal Form

DEFINITION 1.12 **Disjunctive Form** An OR of AND-terms, where each AND-term is an AND of variables or their negations.

DEFINITION 1.13 **Disjunctive Normal Form/DNF** A disjunctive form where each AND-term is an AND of every one of the variables or their negations in turn. You can read a DNF for any propositional formula directly from its truth table.

DEFINITION 1.14 **Conjunctive Form** An AND of OR-terms, where each OR-terms is an OR of variables or their negations.

DEFINITION 1.15 **Conjunctive Normal Form/CNF** A conjunctive form where each OR-term is an OR of every one of the variables or their negations in turn.

THEOREM 1.16 Every propositional formula is equivalent to both a disjunctive normal form and a conjunctive normal form.

1.1.4 The Algebra of Propositions

DEFINITION 1.17 **Propositional Equivalence Axioms**

$$\neg\neg P \equiv P, \quad \text{(double negation)} \tag{1.6}$$
$$P \wedge Q \equiv Q \wedge P, \quad \text{(commutativity of AND)} \tag{1.7}$$
$$P \vee Q \equiv Q \vee P, \quad \text{(commutativity of OR)} \tag{1.8}$$
$$(P \wedge Q) \wedge R \equiv P \wedge (Q \wedge R), \quad \text{(associativity of AND)} \tag{1.9}$$
$$(P \vee Q) \vee R \equiv P \vee (Q \vee R), \quad \text{(associativity of OR)} \tag{1.10}$$
$$\mathrm{T} \wedge P \equiv P, \quad \text{(identity for AND)} \tag{1.11}$$
$$\mathrm{F} \vee P \equiv P, \quad \text{(identity for OR)} \tag{1.12}$$
$$\mathrm{F} \wedge P \equiv \mathrm{F}, \quad \text{(zero for AND)} \tag{1.13}$$
$$\mathrm{T} \vee P \equiv \mathrm{T}, \quad \text{(zero for OR)} \tag{1.14}$$
$$P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R), \quad \text{(distributivity of AND over OR)} \tag{1.15}$$
$$P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R), \quad \text{(distributivity of OR over AND)} \tag{1.16}$$
$$P \wedge P \equiv P, \quad \text{(idempotence for AND)} \tag{1.17}$$
$$P \vee P \equiv P, \quad \text{(idempotence for OR)} \tag{1.18}$$

$$P \wedge \neg P \equiv \mathrm{F}, \qquad \text{(contradiction for AND)} \tag{1.19}$$

$$P \vee \neg P \equiv \mathrm{T}, \qquad \text{(validity for OR)} \tag{1.20}$$

$$\neg(P \wedge Q) \equiv \neg P \wedge \neg Q, \qquad \text{(DeMorgan for AND)} \tag{1.21}$$

$$\neg(P \vee Q) \equiv \neg P \vee \neg Q. \qquad \text{(DeMorgan for OR)} \tag{1.22}$$

THEOREM 1.18 Any propositional formula can be transformed into disjunctive normal form or a conjunctive normal form using the equivalences listed above.

THEOREM 1.19 **Completeness of the Propositional Equivalence Axioms** Two propositional formula are equivalent iff they can be proved equivalent using the equivalence axioms listed above.

1.1.5 Equivalence, Validity, and Satisfiability

DEFINITION 1.20 **Equivalence** Two propositional formulas are equivalent exactly when they have the same truth values in all environments.

Two ways to prove equivalence

- By truth table.
- By propositional equivalent axioms. Convert them both to disjunctive normal form. Then use commutativity to sort the variables and AND-terms so they all appear in some standard order. We claim the formulas are equivalent iff they have the same sorted disjunctive normal form. This is because the way we read off a disjunctive normal form from a truth table shows that two different sorted DNF's over the same set of variables correspond to different truth tables and hence to inequivalent formulas.

These two methods involve essentially the same effort. There is no guarantee that applying the axioms will generally be any easier than using truth tables. No efficient method for verifying validity is known.

THEOREM 1.21 An implication and its contrapositive are equivalent. In contrast, an implication is not equivalent to its converse.

$$P \Rightarrow Q \equiv \neg Q \Rightarrow \neg P, \tag{1.23}$$

$$P \Rightarrow Q \not\equiv Q \Rightarrow P. \tag{1.24}$$

DEFINITION 1.22 **Validity** A valid formula is one which is always true, no matter what environment its variables may have. Valid formulas can be interpreted as the fundamental logical truths.

THEOREM 1.23 **Relationship Between Equivalence and Validity** Equivalence of formulas is really a special case of validity, namely,

$$P \equiv Q \text{ iff } P \Leftrightarrow Q \text{ is valid.} \tag{1.25}$$

Validity can also be viewed as an aspect of equivalence, namely,

$$P \text{ is valid iff } P \equiv \mathrm{T}. \tag{1.26}$$

DEFINITION 1.24 **Satisfiability** A satisfiable formula is one which can sometimes be true — that is, there is some environment of its variables that makes it true.

THEOREM 1.25 **Relationship Between Validity and Satisfiability**

$$P \text{ is satisfiable iff } \neg P \text{ is not valid.} \tag{1.27}$$

DEFINITION 1.26 **SAT** The general problem of deciding whether a proposition is satisfiable.

In the case of system specifications, the AND of all the specifications must be satisfiable.

One approach to SAT is to construct a truth table and check whether or not a T ever appears, but the truth tables grow exponentially with the number of variables. The situation is the same for validity checking, since you can check for validity by checking for satisfiability of a negated formula.

So no one has a good idea how to solve SAT in polynomial time, or how to prove that it cannot be done. The problem of determining whether or not SAT has a polynomial time solution is known as the "P vs. NP" problem, where P stands for problems whose instances can be solved in polynomial time, and NP stands for nondeterministic polynomial time.

1.2 Predicate Formulas

DEFINITION 1.27 **Predicate** A proposition whose truth depends on the value of one or more variables.

1.2.1 Quantifiers

DEFINITION 1.28 **Universal Quantification** An assertion that a predicate is always true.

DEFINITION 1.29 **Existential Quantification** An assertion that a predicate is sometimes true.

DEFINITION 1.30 **Domain/Domain of Discourse** The unnamed nonempty set where all the variables in a formula are understood to range over.

THEOREM 1.31 **De Morgan's Laws for Quantifiers** Moving a NOT across a quantifier changes the kind of quantifier.

$$\neg(\forall x.P(x)) \equiv \exists x.\neg P(x)\,, \tag{1.28}$$

$$\neg(\exists x.P(x)) \equiv \forall x.\neg P(x)\,. \tag{1.29}$$

Quantifiers of the same type ($\forall$ or $\exists$) can be reordered without altering the meaning of the statement. Swapping the order of different kinds of quantifiers usually changes the meaning of a proposition.

THEOREM 1.32

$$\exists x, \forall y . P(x, y) \Rightarrow \forall y, \exists x . P(x, y) . \tag{1.30}$$

1.2.2 Validity for Predicate Formulas

DEFINITION 1.33 **Validity for Predicate Formulas** A formula evaluated to true no matter what the domain of discourse may be, no matter what values its variables may take over the domain, and no matter what interpretations its predicate variables may be given.

DEFINITION 1.34 **Undecidable** An undecidable problem is a decision problem for which it is known to be impossible to construct a single algorithm that always leads to a correct yes-or-no answer. A decision problem is any arbitrary yes-or-no question on an infinite set of inputs.

THEOREM 1.35 **Validity is Undecidable** There is no procedure to determine whether a quantified formula is valid (in contrast to propositional formulas where we can use truth table).

2 Proofs

2.1 Proofs Basis

2.1.1 General Proofs

DEFINITION 2.1 **(General) Proof** A method of establishing/ascertaining/verifying truth.

Truth sometime depends on the eye of the beholder, and what constitutes a proof differs among fields. See Tab. 2.1. Proofs are used in computer science in

- genuine understanding: important results cannot be fully understand until their proofs are understood, and
- certifying that software and hardware will *always* behave correctly, something that no amount of testing can do.

The most important skill, in some ways, is the ability to distinguish a very plausible argument that might not be totally right from a proof which is totally right.

DEFINITION 2.2 **Mathematical Proof** A chain of logical deductions/inference rules from a base set of axioms and previously proved statements that concludes with the proposition in question.

2.1.2 Axioms

DEFINITION 2.3 **Axiom** Propositions that are assumed to be true. The key in math is to identify what your assumptions are so people can see them. Axioms can be contradictory in different contexts.

Axioms should be

Table 2.1
Truth and method to establishing truth.

Field	Truth	Method to establishing truth
Judicial systems	Legal truth	Decided by jury based on the allowable evidence presented at trial
Business	Authoritative truth	Specified by a trusted person or organization
Physics or Biology	Scientific truth	Experiment[a]
Statistics	Probable truth	Statistical analysis of sample data
Philosophy	Philosophical truth	Exposition and persuasion based on some small, plausible arguments
Math	Mathematical truth	Mathematical proof

a Actually, only scientific *falsehood* can be demonstrated by an experiment — when the experiment fails to behave as predicted. But no amount of experiment can confirm that the *next* experiment will not fail. For this reason, scientists rarely speak of truth, but rather of *theories* that accurately predict past, and anticipated future, experiments.

- **consistent**: no proposition can be proved to be both true or false, and
- **complete**: they can be used to prove every proposition whether it is true or false.

THEOREM 2.4 **Gödel's Completeness Theorem** Only need to know a few axioms and rules to prove all valid formulas.

In fact, just a handful of axioms, called the **Zermelo-Fraenkel with Choice axioms (ZFC)**, together with a few logical deduction rules, appear to be sufficient to derive essentially all of mathematics.

THEOREM 2.5 **Gödel's Incompleteness Theorem** It is not possible that there exists any set of axioms that are both consistent and complete. If you want consistent, there will be true facts that you will never be able to prove.

2.1.3 The Axiomatic Method

DEFINITION 2.6 **Axiomatic Method** Starting from a base set of axioms, we established the truth of many additional propositions by providing mathematical proofs. The Axiomatic Method is the standard procedure for establishing truth in mathematics.

DEFINITION 2.7 **Theorem** Important true proposition.

DEFINITION 2.8 **Lemma** A preliminary proposition useful for proving later propositions.

DEFINITION 2.9 **Corollary** A proposition that follows in just a few logical steps from a theorem.

2.1.4 Logical Deductions

DEFINITION 2.10 **Logical Deduction Rules** Rules to prove new propositions using previously proved ones. They are written as: when the **antecedents** (statements above the line) are proved, then we can consider the **conclusion/consequent** (statement below the line) to also be proved.

DEFINITION 2.11 **Sound** Environment of truth variables that makes all the antecedents true must also make the consequent true. Deduction rules must be sound, so if we start off with true axioms and apply sound inference rules, everything we prove will also be true.

THEOREM 2.12 **Relationship between Soundness and Validity** A rule is sound iff

$$\wedge\{\text{antecedents}\} \Rightarrow \text{conclusion} \tag{2.1}$$

is valid.

THEOREM 2.13 The following deduction rules are sound.

$$\frac{P, P \Rightarrow Q}{Q}, \tag{2.2}$$

$$\frac{P \Rightarrow Q, Q \Rightarrow R}{P \Rightarrow R}, \tag{2.3}$$

$$\frac{\neg P \Rightarrow \neg Q}{Q \Rightarrow P}, \tag{2.4}$$

where the first rule is called **modus ponens**.

Proof. By truth table. □

2.2 Proving Propositional Formulas

2.2.1 Proving P

2.2.1.1 By Cases

Reasoning by cases can break a complicated problem into easier subproblems.

- The proof is by case analysis.
- There are 2 cases: . . . At least one of these cases must hold.
- Case 1: . . . This implies that the theorem holds in case 1.
- Case 2: . . . This implies that the theorem holds in case 2, and therefore holds in all cases.

THEOREM 2.14 We call a group a *club* if every pair of people in the group has met, and a group of *strangers* if every pair of people in the group has not met. Then every collection of 6 people includes a club of 3 people or a group of 3 strangers.

Proof. The proof is by case analysis. Let x denote one of the 6 people. There are two cases

1. Among 5 other people besides x, ≥ 3 people have met x.
2. Among 5 other people besides x, ≥ 3 people have not met x.

Since we have split the 5 people into two groups, those who have met with x and those who have not, one of the groups must have ≥ 3 people. Therefore, at least one of these two cases must hold.

Case 1: Suppose that ≥ 3 people have met x. This case splits into two subcases:

Case 1.1: No pair among those people met each other. Then these people are a group of ≥ 3 strangers.

Case 1.2: Some pair among those people have met each other. Then that pair, together with x, form people are a club of ≥ 3 people.

This implies that the theorem holds in Case 1.

Case 2: Suppose that ≥ 3 people have not met x. This case splits into two subcases:

Case 1.1: Every pair among those people have met each other. Then these people are a club of ≥ 3 people.

Case 1.2: Some pair among those people have not met each other. Then that pair, together with x, form people are a group of ≥ 3 strangers.

This implies that the theorem holds in Case 2, and therefore holds in all cases. □

THEOREM 2.15 **Ramsey' Theorem** For any k, every group of $\geq R(k)$ people will include either a size-k club, or a group of size-k strangers. It turns out $R(3) = 6$, $R(4) = 18$, and $R(5)$ is unknown. Paul Erdös considered finding $R(6)$ a hopeless challenge.

2.2.1.2 By Contradiction

If you show that $\neg P \Rightarrow$ F is true, then the only way this is true is $\neg P$ is false, namely P is true.

- We use proof by contradiction.
- Assume the theorem is false, namely $\neg P$.
- Deduce something known to be false.
- This is a contradiction. Therefore, P must be true.

LEMMA 2.16

$$2 \mid n^2 \Rightarrow 2 \mid n\,. \tag{2.5}$$

Proof. We prove the contrapositive

$$2 \nmid n \Rightarrow 2 \nmid n^2\,. \tag{2.6}$$

Assume $2 \nmid n^2$.

$$2 \nmid n \Rightarrow n \text{ is odd} \Rightarrow n^2 = n \cdot n \text{ is also odd} \Rightarrow 2 \nmid n^2\,. \tag{2.7}$$

□

LEMMA 2.17

$$\forall a, b \in \mathbb{N}^+, \exists m, n \in \mathbb{Z}.\ \frac{m}{n} = \frac{a}{b} \wedge \gcd(m, n) = 1\,. \tag{2.8}$$

Proof. We use proof by contradiction. Assume the lemma is false, namely,

$$\exists a, b \in \mathbb{N}^+, \forall m, n \in \mathbb{N}^+.\ \frac{m}{n} \neq \frac{a}{b} \vee \gcd(m, n) > 1\,. \tag{2.9}$$

That is to say,

$$\exists a, b \in \mathbb{N}^+, \forall m, n \in \mathbb{N}^+.\ \frac{m}{n} = \frac{a}{b} \Rightarrow \gcd(m, n) > 1\,. \tag{2.10}$$

Define the set C as

$$C := \left\{ m \in \mathbb{N}^+ \mid \exists n \in \mathbb{N}^+ .\ \frac{m}{n} = \frac{a}{b} \right\} . \tag{2.11}$$

Because $a \in C$, $C \neq \varnothing$. By the Well Ordering Principle, there will be a smallest element $m_0 \in C$, and by the definition of C, there is an $n_0 \in \mathbb{N}^+$, such that

$$\frac{m_0}{n_0} = \frac{a}{b} . \tag{2.12}$$

By the assumption,

$$\gcd(m_0, n_0) = d > 1 . \tag{2.13}$$

Therefore

$$\frac{\frac{m_0}{d}}{\frac{n_0}{d}} = \frac{m_0}{n_0} = \frac{a}{b} , \tag{2.14}$$

It implies that $\frac{m_0}{d} \in C$. This contradicts the assumption that m_0 is the smallest element in C. Thus, the lemma must be true. □

THEOREM 2.18 $\sqrt{2}$ is irrational.

Proof. We use proof by contradiction. Suppose the theorem is false, namely, $\sqrt{2}$ is rational. Then

$$\exists m, n \in \mathbb{N}^+ .\ \sqrt{2} = \frac{m}{n} \wedge \gcd(m, n) = 1 . \tag{2.15}$$

Squaring both sides of $\sqrt{2} = \frac{m}{n}$ gives

$$2 = \frac{m^2}{n^2} \Rightarrow m^2 = 2n^2 \Rightarrow 2 \mid m^2 \Rightarrow 2 \mid m \Rightarrow \exists d \in \mathbb{Z} .\ m = 2d , \tag{2.16}$$

Squaring both sides of $2d = m$ gives

$$4d^2 = m^2 = 2n^2 \Rightarrow n^2 = 2d^2 \Rightarrow 2 \mid n^2 \Rightarrow 2 \mid n . \tag{2.17}$$

$$2 \mid m \wedge 2 \mid n \Rightarrow \gcd(m, n) \geq 2 , \tag{2.18}$$

which contradicts the fact that $\gcd(m, n) = 1$. Thus $\sqrt{2}$ must be irrational. □

2.2.2 Proving $P \Rightarrow Q$

2.2.2.1 Direct

- Assume P.
- Show that Q logically follows.

THEOREM 2.19

$$0 \leq x \leq 2 \Rightarrow -x^3 + 4x + 1 > 0\,. \tag{2.19}$$

Proof. Assume $0 \leq x \leq 2$. Then

$$x \geq 0 \wedge 2 - x \geq 0 \wedge 2 + x \geq 0\,. \tag{2.20}$$

A product of these terms is also nonnegative,

$$x(2-x)(2+x) \geq 0\,. \tag{2.21}$$

Adding 1 to this product gives a positive number,

$$x(2-x)(2+x) + 1 = -x^3 + 4x + 1 > 0 \tag{2.22}$$

as claimed. □

2.2.2.2 Prove $\neg Q \Rightarrow \neg P$

- We prove the contrapositive: $\neg Q \Rightarrow \neg P$.
- Assume $\neg Q$.
- Show that $\neg P$ logically follows.

THEOREM 2.20

$$r \text{ is irrational} \Rightarrow \sqrt{r} \text{ is irrational.} \tag{2.23}$$

Proof. We prove the contrapositive:

$$\sqrt{r} \text{ is rational} \Rightarrow r \text{ is rational.} \tag{2.24}$$

Assume $\sqrt{r}$ is rational. Then

$$\exists m, n \in \mathbb{Z}.\ \sqrt{r} = \frac{m}{n}\,. \tag{2.25}$$

Squaring both sides gives

$$r = \frac{m^2}{n^2}\,. \tag{2.26}$$

Since $m^2, n^2 \in \mathbb{Z}$, r is also rational. □

2.2.3 Proving $P \Leftrightarrow Q$

2.2.3.1 Prove $P \Rightarrow Q \wedge Q \Rightarrow P$

- We prove $P \Rightarrow Q$ and vice-versa.
- First, we show that $P \Rightarrow Q$.
- Now, we show that $Q \Rightarrow P$.

2.2.3.2 Construct a Chain of IFFs

- We construct a chain of iff implications. Starting with *P*.
- Prove P holds iff a second statement holds, which holds iff a third statement holds, and so forth until you reach Q.

THEOREM 2.21 For a sequence of values $(x_i)_{i=1}^n$, let

$$\mu := \frac{1}{n}\sum_{i=1}^{n} x_i\,, \tag{2.27}$$

$$\sigma := \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^2}\,. \tag{2.28}$$

Then

$$\sigma = 0 \Leftrightarrow \forall i \in [1,n]\,.\, x_i = \mu\,. \tag{2.29}$$

Proof. We construct a chain of iff implications. Starting with $\sigma = 0$. Since 0 is the only number whose square root is 0,

$$\sigma = 0 \Leftrightarrow \sum_{i=1}^{n}(x_i-\mu)^2 = 0\,. \tag{2.30}$$

Squares of real numbers are always nonnegative, so every term is nonnegative,

$$\sum_{i=1}^{n}(x_i-\mu)^2 = 0 \Leftrightarrow \forall i \in [1,n]\,.\,(x_i-\mu)^2 = 0\,. \tag{2.31}$$

But a term $(x_i-\mu)^2 = 0$ iff $x_i = \mu$, so

$$\forall i \in [1,n]\,.\,(x_i-\mu)^2 = 0 \Leftrightarrow \forall i \in [1,n]\,.\,x_i = \mu\,. \tag{2.32}$$

□

2.3 Proving $\forall n \in \mathbb{N}.P(n)$

2.3.1 The Well Ordering Principle

2.3.1.1 Well Ordering Proofs

THEOREM 2.22 **The Well Ordering Principle** Every nonempty set of nonnegative integers has a smallest element.

$$\frac{\exists n \in \mathbb{N}.\ P(n)}{(\exists n_0 \in \mathbb{N}.\ P(n_0)) \wedge (\forall n \in \mathbb{N}.\ P(n) \Rightarrow n \geq n_0)}. \tag{2.33}$$

- We use proof by contradiction.
- Define the set, C, of counterexamples to P being true. Specifically,

$$C := \{n \in \mathbb{N} \mid \neg P(n)\}. \tag{2.34}$$

- Assume for proof by contradiction that $C \neq \varnothing$.
- By the Well Ordering Principle, there will be a smallest element, n_0, in C.
- Reach a contradiction somehow — often by showing that $P(n_0)$ is actually true or by showing that there is another member of C that is smaller than n_0.
- This is a contradiction. Therefore $C = \varnothing$, that is, no counterexamples exist, namely P must be true.

THEOREM 2.23 For any positive integer $n \geq 1$, n can be factored as a product of primes.

Proof. We use proof by contradiction. Define the set, C, of counterexamples to the theorem being true. Specifically,

$$C := \{n > 1 \mid n \text{ cannot be factored as a product of primes}\}. \tag{2.35}$$

Assume for proof by contradiction that $C \neq \varnothing$.

By the Well Ordering Principle, there will be a smallest element, n_0, in C. n_0 cannot be prime, because a prime by itself is considered a length one product of primes. So n_0 must be a product of two integers a and b where $1 < a, b < n$, and so $a, b \notin C$. In other words, a can be written as a product of primes $\prod_{i=1}^{k} p_i$, and b can be written as a product of primes $\prod_{j=1}^{l} q_j$. Therefore

$$n_0 = ab = \prod_{i=1}^{k} p_i \prod_{j=1}^{l} q_j \tag{2.36}$$

can be written as a product of primes, contradicting to the claim that $n_0 \in C$. Therefore, $C = \varnothing$, that is, no counterexamples exists, namely, the theorem must be true. □

2.3.1.2 Well Ordered Sets

DEFINITION 2.24 **Well Ordered Set** A set A is well ordered iff every nonempty subset $S \subseteq A$ has a minimum element.

THEOREM 2.25 The following sets are well ordered.
- every finite set,
- $\mathbb{N}$,
- $r\mathbb{N}$, where $r \in \mathbb{R}^+$,
- $\mathbb{N} - k := \{i - k \mid i \in \mathbb{N}\}$, where $k \in \mathbb{N}$,
- $F := \{\frac{i}{i+1} \mid i \in \mathbb{N}\}$,
- $\mathbb{N} + F := \{i + f \mid i \in \mathbb{N} \wedge f \in F\}$,
- $\mathbb{N} \cup F$.

Proof. $\mathbb{N}$ is well ordered by the Well Ordering Principle: Every nonempty subset $S \subseteq \mathbb{N}$ has a smallest element.

$\mathbb{N} - k$ is well ordered since, for any nonempty subset $A \in (\mathbb{N} - k)$, $A + k \in \mathbb{N}$, and because $\mathbb{N}$ is well ordered, $A + k$ has a minimum element α. Then it is easy to see that $\alpha - k$ is the minimum element of A.

F is well ordered since, the minimum element of any nonempty subset $A \in F$ is simply the one with the minimum numerator when expressed in the form $\frac{i}{i+1}$.

$\mathbb{N} + F$ is well ordered since, for any nonempty subset $A \in (\mathbb{N} + F)$, look at $\{i \in \mathbb{N} \mid \exists f \in F.\ i + f \in A\}$. This is a nonempty set of $\mathbb{N}$, and $\mathbb{N}$ is well ordered, so there must be a minimum element i_0. Consider $\{f \in F \mid i_0 + f \in A\}$, this is a nonempty set of F, and F is well ordered, so there must be a minimum element f_0. Therefore, $i_0 + f_0$ is the minimum element of A.

$\mathbb{N} \cup F$ is well ordered since, for any nonempty subset $A := (\mathbb{N}' \cup F') \subseteq (\mathbb{N} \cup F)$, if $F' \neq \varnothing$, $\min A = \min F'$; if $F' = \varnothing$, $\min A = \min \mathbb{N}'$. □

THEOREM 2.26 The set $\{x \in \mathbb{R} \mid x \geq 0\}$ is not well ordered.

Proof. For example, the subset $\{x \in \mathbb{R} \mid x > 0\}$ does not have the minimum element. □

COROLLARY 2.27 Any set A of integers with a lower bound b is well ordered. That is to say, any nonempty set of integers with an lower bound has a minimum element.

Proof. $\mathbb{N} - \lfloor b \rfloor$ is well ordered, and any subset $S \subseteq A$ is also a subset $S \subseteq (\mathbb{N} - \lfloor b \rfloor)$ and so has a minimum. Therefore, A is well ordered. □

COROLLARY 2.28 Any nonempty set of integers A with an upper bound b has a maximum element.

Proof. b is an upper bound of A, so $-b$ is a lower bound of $-A$. By Corollary 2.27, $-A$ has a minimum element $-\alpha$, so α is a maximum element of A. □

2.3.2 Ordinary Induction

THEOREM 2.29 **The Induction Principle**

$$\frac{P(0), \forall n \in \mathbb{N}.\ P(n) \Rightarrow P(n+1)}{\forall n \in \mathbb{N}.\ P(n)}. \tag{2.37}$$

- We use proof by induction.
- The induction hypothesis, $P(n)$, will be ...
- Base case: $P(0)$ is true, because ...
- Inductive step: Assume that $P(n)$ is true ... which proves $P(n+1)$.
- So it follows by induction that $P(n)$ is true for all $n \in \mathbb{N}$.

When an induction proof will not go through, a good move is to use a strong induction hypothesis, which implies your previous hypothesis. But the stronger assertion must actually be true.

2.3.3 Strong Induction

THEOREM 2.30 **The Strong Induction Principle**

$$\frac{P(0), \forall n \in \mathbb{N}.\ (P(0) \wedge P(1) \wedge \cdots \wedge P(n)) \Rightarrow P(n+1)}{\forall n \in \mathbb{N}.\ P(n)}. \tag{2.38}$$

Or it can be written as

$$\frac{P(0), \forall n \in \mathbb{N}.\ (\forall k \leq n \in \mathbb{N}.P(k)) \Rightarrow P(n+1)}{\forall n \in \mathbb{N}.\ P(n)}. \tag{2.39}$$

- We use proof by strong induction.
- The induction hypothesis, $P(n)$, will be ...
- Base case: $P(0)$ is true, because ...
- Inductive step: Assume that $P(0), P(1), \ldots, P(n)$ together is true ... which proves $P(n+1)$.
- So it follows by induction that $P(n)$ is true for all $n \in \mathbb{N}$.

2.3.4 Strong Induction vs. Induction vs. Well Ordering

Strong induction has the same power of ordinary induction.

- Ordinary induction is a special case of strong induction.
- Any strong induction proof can automatically be reformatted into an ordinary induction proof. Define $Q(n) := \forall k \leq n \in \mathbb{N}.\ P(k)$. The base case is the same, and the inductive step: $Q(n) \Rightarrow Q(n+1)$.

Any well ordering proof can automatically be reformatted into and from an induction proof.

Therefore, all the three proof methods are simply different formats for presenting the same mathematical reasoning.

3 State Machines

3.1 State and Transitions

DEFINITION 3.1 **State Machine** A binary relation (transition relation) on a set of **states**. An arrow in the graph of the transition relation is called a **transition**. A state machine also comes equipped with a designated **start state**. The transition relation is also called the **state graph** of the machine. More formally, we can define a state machine as

- A set of states: S.
- A start state: $s_0 \in S$.
- A set of allowed transitions between states: $\{(r, s) \in S \times S \mid r \to s\}$.

State machines are a simple, abstract model of step-by-step processes.

DEFINITION 3.2 **Execution** An execution of the state machine is a (possibly infinite) sequence of states with the property that

- It begins with the start state.
- If r and s are consecutive states in the sequence, then $r \to s$.

A state machine execution describes a possible sequence of steps a machine might take.

DEFINITION 3.3 **Reachable State** A state is called reachable if it appears in some execution.

DEFINITION 3.4 **Deterministic State Machine** A state machine where at most one transition out of each state.

DEFINITION 3.5 **Nondeterministic State Machine** A state machine where some states have transitions to several different states.

3.2 The Invariant Principle

DEFINITION 3.6 **Preserved Invariant** A preserved invariant of a state machine is a predicate, P, on states, such that whenever $r \to s$, $P(r) \Rightarrow P(s)$. A preserved invariant is a property that is preserved through a series of operations or steps.

THEOREM 3.7 **The Invariant Principle** If a preserved invariant of a state machine is true for the start state, then it is true for all reachable states.

Proving "In every reachable state, the property ... holds"

- By using invariant principle.
- Let $P(s)$ be ...
- $P(s_0)$ is true in the start state s_0, because ...

- We need to show that $P(s)$ is a preserved invariant, namely, for any transition $r \to s$, $P(r) \Rightarrow (s)$. Assume that $P(r)$ is true ... which proves $P(s)$.
- So it follows by invariant principle that $P(s)$ is true for all reachable state s.

DEFINITION 3.8 **Partial Correctness** The final results, if any, of the process must satisfy system requirements. This is often proved by invariant principle.

3.3 Termination

DEFINITION 3.9 **Termination** The process does always produce some final value.

DEFINITION 3.10 **Derived Variables** Like potential functions in physics, derived variables are value assignments for states. ϕ: $S \mapsto \mathbb{R}$.

DEFINITION 3.11 $\mathbb{N}$**-valued Derived Variables** Derived variables whose values range over $\mathbb{N}$.

DEFINITION 3.12 **Strictly Decreasing Derived Variables** A derived variable ϕ: $S \mapsto \mathbb{R}$ is strictly decreasing iff

$$r \to s \Rightarrow \phi(r) < \phi(s)\,. \tag{3.1}$$

DEFINITION 3.13 **Weakly Decreasing Derived Variables** A derived variable ϕ: $S \mapsto \mathbb{R}$ is weakly decreasing iff

$$r \to s \Rightarrow \phi(r) \leq \phi(s)\,. \tag{3.2}$$

THEOREM 3.14 If ϕ is a strictly decreasing $\mathbb{N}$-valued derived variable of a state machine, then the length of any execution starting at state s is $\leq \phi(s)$.

LEMMA 3.15 A set of numbers is well ordered iff it has no infinite decreasing sequences.

THEOREM 3.16 If there exists a strictly decreasing derived variable whose range is a well ordered set, then every execution terminates.

A weakly decreasing derived variable does not guarantee that every execution terminates.

3.4 Fast Exponentiation

The most straightforward way to compute a^b is to multiply a by itself $b-1$ times, which takes $O(b)$ steps. But the solution can be found in considerably fewer multiplications by Fast Exponentiation, see Alg. 1. We can model this algorithm by a state machine.

- states $S := \mathbb{R} \times \mathbb{R} \times \mathbb{N}$,
- start state $(x_0, y_0, z_0) := (a, 1, b)$,
- transitions:

$$(x, y, z) \to \begin{cases} (x^2, y, \text{qcnt}(z, 2)) & \text{if } z \neq 0 \wedge z \text{ is even;} \\ (x^2, xy, \text{qcnt}(z, 2)) & \text{if } z \neq 0 \wedge z \text{ is odd.} \end{cases} \tag{3.3}$$

LEMMA 3.17 The Fast Exponentiation algorithm will terminate in $O(\lg b)$ steps.

Proof. z is initialed by b, and gets at least halved with each transition. So it cannot be halved more than $\lceil \lg b \rceil + 1$ times before hitting 0 and causing the algorithm to terminate. □

LEMMA 3.18 The preserved invariant of Fast Exponentiation is

$$P((x, y, z)) := z \in \mathbb{N} \wedge yx^z = a^b . \tag{3.4}$$

Proof. For a transition $(x, y, z) \to (u, v, w)$, assume $P((x, y, z))$ holds. Since there is a transition from (x, y, z), we have $z \neq 0$. We consider two cases.
Case 1: if z is even, then

$$(u, v, w) = (x^2, y, \text{qcnt}(z, 2)) = \left(x^2, y, \frac{z}{2}\right) . \tag{3.5}$$

Therefore, $w \in \mathbb{N}$ and

$$vu^w = y(x^2)^{\frac{z}{2}} = yx^{\frac{2z}{2}} = yx^z = a^b . \tag{3.6}$$

Case 2: if z is odd, then

$$(u, v, w) = (x^2, xy, \text{qcnt}(z, 2)) = \left(x^2, xy, \frac{z-1}{2}\right) . \tag{3.7}$$

Therefore, $w \in \mathbb{N}$ and

$$vu^w = xy(x^2)^{\frac{z-1}{2}} = yx^{1+\frac{2(z-1)}{2}} = yx^z = a^b . \tag{3.8}$$

So in both cases, $P((u, v, w))$ holds, proving that P is a preserved invariant. □

THEOREM 3.19 $P((x, y, z))$ is a partial correctness.

Proof. By using invariant principle.
$P((x_0, y_0, z_0))$ is true in the start state $(x_0, y_0, z_0) = (a, 1, b)$, because $b \in \mathbb{N}$ and

$$y_0 x_0^{z_0} = a^b . \tag{3.9}$$

Algorithm 1 Fast Exponentiation.

Input: $a \in \mathbb{R}$; $b \in \mathbb{N}$.
Output: a^b.

$x \leftarrow a$
$y \leftarrow 1$
$z \leftarrow b$
while $z \neq 0$
  $r \leftarrow \text{rem}(z, 2)$
  $z \leftarrow \text{qcnt}(z, 2)$
  if $r = 1$
    $y \leftarrow xy$
  $x \leftarrow x^2$
return y

Since $P((x, y, z))$ is a preserved invariant, So it follows by invariant principle that $P((x, y, z))$ is true for all reachable state (x, y, z). Therefore, when the algorithm terminates at $(x, y, 0)$,

$$yx^0 = y = a^b . \tag{3.10}$$

□

4 Mathematical Data Types

4.1 Sets

4.1.1 Set Basis

DEFINITION 4.1 **Set** A set is an unordered collection of objects, which are called the **elements** of the set.

Two ways to describe a set

- List the elements explicitly inside braces.
- Use set builder notation: The set consists of all values in A that make the predicate true: $\{x \in A \mid P(x)\}$.

The elements of the set do not have to be in the same type; Sets can contain sets; Any object is, or is not, an element of a given set; A set cannot contain the same object more than once.

DEFINITION 4.2 **Venn Diagram** A graphical picture in which sets are represented as regions of the plane.

DEFINITION 4.3 **n-Set** A finite set of n elements.

DEFINITION 4.4 **Singleton** A 1-set.

DEFINITION 4.5 **k-Subset** A subset of k elements of a set.

4.1.2 Comparing and Combining Sets

DEFINITION 4.6 **Equal** $=$

$$A = B := \forall x.\, x \in A \Leftrightarrow x \in B\,. \tag{4.1}$$

DEFINITION 4.7 **Subset** $\subseteq$

$$A \subseteq B := \forall x.\, x \in A \Rightarrow x \in B \equiv A \cap \bar{B} = \varnothing\,. \tag{4.2}$$

DEFINITION 4.8 **Strict Subset/Proper Subset** $\subset$

$$A \subset B := A \subseteq B \wedge A \neq B\,. \tag{4.3}$$

DEFINITION 4.9 **Union** $\cup$

$$A \cup B := \{x \mid x \in A \vee x \in B\}\,. \tag{4.4}$$

DEFINITION 4.10 **Intersection** $\cap$

$$A \cap B := \{x \mid x \in A \wedge x \in B\}\,. \tag{4.5}$$

DEFINITION 4.11 **Set Difference** $-$

$$A - B := \{x \mid x \in A \wedge x \notin B\}. \tag{4.6}$$

DEFINITION 4.12 **Universe** U All the sets under consideration are subsets of some larger set U called universe.

DEFINITION 4.13 **Complement** $\bar{A}$

$$\bar{A} := \{x \mid x \notin A\}. \tag{4.7}$$

DEFINITION 4.14 **Power Set** $\mathrm{pow}\, A$

$$\mathrm{pow}\, A := \{S \mid S \subseteq A\}. \tag{4.8}$$

DEFINITION 4.15 **Disjoint** Two set A and B are disjoint if they have no elements in common, that is, if $A \cap B = \varnothing$.

DEFINITION 4.16 **Partition** A collection of $A := \{A_i\}_{i=1}^n$ of nonempty sets forms a partition of a set A if

- The sets are pairwise disjoint, that is

$$\forall i \neq j.\, A_i \cap A_j = \varnothing. \tag{4.9}$$

- Their union is A, that is

$$\bigcap_{i=1}^{n} A_i = A. \tag{4.10}$$

In other words, $\{A\}_{i=1}^n$ forms a partition of A if each element of A appears in exactly one $A_i \in A$. Each set A_i is called a **block** of the partition.

4.1.3 Laws of Set Operations

We can prove a set equality involving the basic set operations by checking that a corresponding propositional formula is valid.

THEOREM 4.17 **Empty Set Laws**

$$A \cap \varnothing = \varnothing, \tag{4.11}$$

$$A \cup \varnothing = A. \tag{4.12}$$

THEOREM 4.18 **Idempotency Laws**

$$A \cap A = A, \tag{4.13}$$

$$A \cup A = A. \tag{4.14}$$

THEOREM 4.19 **Commutative Laws**

$$A \cap B = B \cap A\,, \tag{4.15}$$
$$A \cup B = B \cup A\,. \tag{4.16}$$

THEOREM 4.20 **Associative Laws**

$$A \cap (B \cap C) = (A \cap B) \cap C\,, \tag{4.17}$$
$$A \cup (B \cup C) = (A \cup B) \cup C\,. \tag{4.18}$$

THEOREM 4.21 **Distributive Laws**

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)\,, \tag{4.19}$$
$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)\,. \tag{4.20}$$

Proof. The theorem is equivalent to the proposition that

$$\forall x.\, x \in A \cap (B \cup C) \Leftrightarrow x \in (A \cap B) \cup (A \cap C)\,. \tag{4.21}$$

We construct a chain of iff implications.

$$x \in A \cap (B \cup C) \Leftrightarrow (x \in A) \wedge (x \in B \cup C) \tag{4.22}$$
$$\Leftrightarrow (x \in A) \wedge (x \in B \vee x \in C) \tag{4.23}$$
$$\Leftrightarrow (x \in A \wedge x \in B) \vee (x \in A \wedge x \in C) \tag{4.24}$$
$$\Leftrightarrow (x \in A \cap B) \vee (x \in A \cap C) \tag{4.25}$$
$$\Leftrightarrow x \in (A \cap B) \cup (A \cap C)\,. \tag{4.26}$$

□

THEOREM 4.22 **Absorption Laws**

$$A \cap (A \cup B) = A\,, \tag{4.27}$$
$$A \cup (A \cap B) = A\,. \tag{4.28}$$

THEOREM 4.23 **Complement Laws**

$$\bar{\bar{A}} = A\,, \tag{4.29}$$
$$A \cap \bar{A} = \varnothing\,, \tag{4.30}$$
$$A \cup \bar{A} = U\,. \tag{4.31}$$

THEOREM 4.24 **De Morgan's Laws**

$$A - (B \cap C) = (A - B) \cup (A - C)\,, \tag{4.32}$$
$$A - (B \cup C) = (A - B) \cap (A - C)\,. \tag{4.33}$$

COROLLARY 4.25 **De Morgan's Laws with Complements**

$$\overline{B \cap C} = \bar{B} \cup \bar{C}, \tag{4.34}$$
$$\overline{B \cup C} = \bar{B} \cap \bar{C}. \tag{4.35}$$

4.2 Sequences

DEFINITION 4.26 **Sequence** A sequence is an ordered collection of objects, which are called the **components** of the sequence.

Differences between sets and sequences

- The elements of a set are required to be distinct, but elements in a sequence can be the same.
- The elements in a sequence have a specified order, but the elements of a set do not.
- We use $\varnothing$ to denote the empty set, while we use λ to denote the empty sequence.

DEFINITION 4.27 **Cartesian Product**

$$A_1 \times A_2 \times \cdots \times A_n := \{(x_1, x_2, \ldots, x_n) \mid \forall i. x_i \in A_i\}. \tag{4.36}$$

A product of n copies of a set A is denoted as A^n.

DEFINITION 4.28 **n-Tuple** A finite sequence of n components.

DEFINITION 4.29 **Pair** Length two sequences.

4.3 Recursive Data Types

4.3.1 Recursive Definitions and Structural Induction

DEFINITION 4.30 **Recursive Definition** A way to construct new data elements by from previous one. Definitions of recursive data types have two parts:

- Base case(s): specifying that some known mathematical elements are in the data type.
- Constructor case(s): that specify how to construct new data elements from previously constructed elements or from base elements.

THEOREM 4.31 The Structural Induction Principle

Let P be a predicate on a recursively defined data type T. If
- $P(b)$ is true for each base case element $b \in T$.
- for all two-argument constructors c:

$$\forall s, t \in T.\ (P(s) \wedge P(t)) \Rightarrow P(c(s,t))\,. \tag{4.37}$$

and likewise for all constructors taking other numbers of arguments.

Then

$$\forall t \in T.\ P(t)\,. \tag{4.38}$$

Proving that all the elements of a recursively defined data type have some property:
- By structural induction on the recursive definition of ...
- The induction hypothesis P will be ...
- Base case ...
- Constructor case ...
- So it follows by structural induction that $\forall t \in T.\ P(t)$.

DEFINITION 4.32 **Function on Recursive Data Type** The function f on recursive data types can be defined recursively using the same cases as the data type definition. Specifically, define the value of f for the base cases of the data type definition, then define the value of f in each constructor case in terms of the values of f on the component data items.

DEFINITION 4.33 **Ambiguous Recursive Definition** Case when a recursive definition of a data type allows the same element to be constructed in more than one way. Recursively defining a function on an ambiguous data type definition usually will not work.

4.3.2 String

DEFINITION 4.34 **Finite String A^*** Let A be a nonempty set called an **alphabet**, whose elements are referred to as **characters/letters/symbols/digits**. The recursive data type A^* of strings over alphabet A is defined as follows:
- Base case: $\lambda \in A^*$.
- Constructor case: $(c \in A \wedge s \in A^*) \Rightarrow (c,s) \in A^*$.

Define, $\{0,1\}^n$ to be the n-bit binary strings, $\{0,1\}^*$ to be the finite-bit binary strings, and $\{0,1\}^\omega$ to be the infinite-bit binary strings.

DEFINITION 4.35 **Length of a String $|s|$** The length of a string is defined as follows:
- Base case: $|\lambda| := 0$.
- Constructor case: $|(c,s)| := 1 + |s|$.

DEFINITION 4.36 **k-String** A string of length k. We can view a k-string over a set A as an element of the Cartesian product of A^k.

DEFINITION 4.37 **Substring** A substring s' of a string s is an ordered sequence of consecutive elements of s.

DEFINITION 4.38 **k-Substring** A substring of length k.

DEFINITION 4.39 **Concatenation of Strings $t \cdot r$** The concatenation of a string is defined as follows:

- Base case: $\lambda \cdot r := r$.
- Constructor case: $(a, s) \cdot r := (a, s \cdot r)$.

DEFINITION 4.40 **Number Occurrences** The number occurrences of a character $a \in A$ in a string is defined as follows:

- Base case: $\#_a(\lambda) := 0$.
- Constructor case: $\#_a((c, s)) := \mathbb{I}\{c = a\} + \#_c(s)$.

4.3.3 Matched String

DEFINITION 4.41 **Matched String** A string $t \in \{[]\}^*$ is called a matched string if its brackets "match up" in the usual way. The recursive data type, M, of matched string is defined as follows:

- Base case: $\lambda \in M$.
- Constructor case: $s, r \in M \Rightarrow [s]\, r \in M$.

4.3.4 Elementary Single Variable Functions

DEFINITION 4.42 **Elementary Single Variable Functions** The recursive data type, F, of elementary single variable functions is defined as follows:

- Base cases:
  - $\forall c \in \mathbb{R}.\ c \in F$.
  - $x \in F$.
  - $\sin x \in F$.
- Constructor cases: If $f, g \in F$, then
  - $f + g \in F$.
  - $f \cdot g \in F$.
  - $f \circ g \in F$.
  - $2^f \in F$.
  - $f^{-1} \in F$.

THEOREM 4.43 F is closed under derivatives.

$$f \in F \Rightarrow \frac{\mathrm{d}f}{\mathrm{d}x} \in F. \tag{4.39}$$

4.3.5 Arithmetic Expressions

DEFINITION 4.44 **Arithmetic Expression** The recursive data type, E, of arithmetic expression is defined as follows:

- Base cases:
  - $x \in E$.
  - $\forall k \in \mathbb{N}.\ k \in E$.
- Constructor cases: If $s, r \in E$, then
  - $(s + r) \in E$.
  - $(s \cdot r) \in E$.
  - $-(s) \in E$.

E's are fully bracketed.

DEFINITION 4.45 **Evaluation Function** Given any $t \in E$, and $n \in \mathbb{Z}$ for the variable x, we can evaluate t to finds its value $\mathrm{eval}(t, n)$. The evaluation function $\mathrm{eval}\ E \times \mathbb{Z} \mapsto \mathbb{Z}$ is defined as follows:

- Base cases:
  - $\forall n \in \mathbb{Z}.\ \mathrm{eval}(x, n) := n$.
  - $\forall n \in \mathbb{Z}, \forall k \in \mathbb{N}.\ \mathrm{eval}(k, n) := k$.
- Constructor cases: If $s, r \in E$, then
  - $\forall n \in \mathbb{Z}.\ \mathrm{eval}((s + r), n) := \mathrm{eval}(s, n) + \mathrm{eval}(r, n)$.
  - $\forall n \in \mathbb{Z}.\ \mathrm{eval}((s \cdot r), n) := \mathrm{eval}(s, n) \cdot \mathrm{eval}(r, n)$.
  - $\forall n \in \mathbb{Z}.\ \mathrm{eval}(-(s), n) := -\mathrm{eval}(s, n)$.

DEFINITION 4.46 **Substitution Function** Given any $e, t \in E$, we can substitute e for each of the x in t to get the result $\mathrm{subs}(e, t)$. The substitution function $\mathrm{subs}\ E \times E \mapsto E$ is defined as follows:

- Base cases:
  - $\forall e \in E.\ \mathrm{subs}(e, x) := e$.
  - $\forall e \in E, \forall k \in \mathbb{N}.\ \mathrm{subs}(e, k) := k$.
- Constructor cases: If $s, r \in E$, then
  - $\forall e \in E.\ \mathrm{subs}(e, (s + r)) := (\mathrm{subs}(e, s) + \mathrm{subs}(e, r))$.
  - $\forall e \in E.\ \mathrm{subs}((s \cdot r), n) := (\mathrm{subs}(e, s) \cdot \mathrm{subs}(e, r))$.
  - $\forall e \in E.\ \mathrm{subs}(-(s), n) := -(\mathrm{subs}(e, s))$.

If we want to find the value of $\mathrm{subs}(e, t)$ when $x = n$, there are two approaches:

- Substitution model: $\mathrm{eval}(\mathrm{subs}(e, t), n)$. In this case, e may appear multiple times in t, so $\mathrm{eval}(e, n)$ may be computed multiple times.

- Environment model: $\text{eval}(t, \text{eval}(e, n))$. In this case, we only compute $\text{eval}(e, n)$ once.

4.4 Recursive Functions on $\mathbb{N}$

DEFINITION 4.47 **Factorial Function** The factorial function is defined as follows:
- $0! := 1$.
- $\forall n \in \mathbb{N}.\ (n+1)! := (n+1) \cdot n!$.

DEFINITION 4.48 **Fibonacci Numbers** The n-th Fibonacci number is defined as follows:
- $F(0) := 0$.
- $F(1) := 1$.
- $\forall n \in \mathbb{N}.\ F(n+2) = F(n) + F(n+1)$.

DEFINITION 4.49 **Summation Notation** The summation notation is defined as follows:
- $\sum_{i=1}^{0} f(i) := 0$.
- $\forall n \in \mathbb{N}.\ \sum_{i=1}^{n+1} := f(n+1) + \sum_{i=1}^{n} f(i)$.

DEFINITION 4.50 **Product Notation** The product notation is defined as follows:
- $\prod_{i=1}^{0} f(i) := 1$.
- $\forall n \in \mathbb{N}.\ \prod_{i=1}^{n+1} := f(n+1) \cdot \prod_{i=1}^{n} f(i)$.

5 Binary Relations

5.1 Binary Relations

5.1.1 Relation Basis

DEFINITION 5.1 **Binary Relation** A binary relation R on two sets A and B is a subset of the Cartesian product of A and B, namely $R \subseteq A \times B$. We usually denote as $R\colon A \mapsto B$. It is said to be "between A and B", or "from A to B". Set A is called **domain** of A and set B is called **codomain**. When the domain and codomain are the same set A, we simply say the relation is "on A". An n-ary relation on sets $A_1, A_2, \ldots, A_n$ is a subset of $A_1 \times A_2 \times \cdots \times A_n$.

DEFINITION 5.2 **Relation Bipartite Diagram** Every relation $R\colon A \mapsto B$ can be represented as a bipartite graph $G = (V, E)$ where V_L corresponding to the elements of the domain, and V_R corresponding to the elements of the column. There is an edge $\{a, b\} \in E$ iff aRb. Similarly, every bipartite graph determines a relation between the nodes from V_L and the nodes from V_R.

DEFINITION 5.3 **Relation Digraph** Every relation $R\colon A \mapsto A$ can be represented as a digraph $G = (V, E)$ where $V = A$ and $E = R$. Similarly, every digraph determines a relation on V.

DEFINITION 5.4 **Image** The image of a set $S \subseteq A$ under a relation R, is the set of elements of the codomain B of R that are related to some element in S. In terms of the relation diagram, $R(S)$ is the set of points with an arrow coming in that starts from some point in S.

$$R(S) := \{b \in B \mid \exists a \in S. aRb\}. \tag{5.1}$$

DEFINITION 5.5 **Range**

$$\operatorname{ran} R := R(\operatorname{dom} R). \tag{5.2}$$

DEFINITION 5.6 **Composition**

$$a(R \circ S)b := \exists c.\ aSc \wedge cRb. \tag{5.3}$$

DEFINITION 5.7 **Product**

$$(a_1, a_2)(R \times S)(b_1, b_2) := a_1 R b_1 \wedge a_2 S b_2. \tag{5.4}$$

Where

$$\operatorname{dom}(R \times S) = \operatorname{dom} R \times \operatorname{dom} S, \tag{5.5}$$

$$\operatorname{cod}(R \times S) = \operatorname{cod} R \times \operatorname{cod} S, \tag{5.6}$$

$$\tag{5.7}$$

DEFINITION 5.8 **Inverse** A relation from B to A. R^{-1}: $B \mapsto A$. R^{-1} is the relation you get by reversing the direction of the arrows in the diagram of R.

$$bR^{-1}a := aRb\,. \tag{5.8}$$

DEFINITION 5.9 **Inverse Image** The image of a set under the relation R^{-1}.

$$R^{-1}(S) := \{a \in A \mid \exists b \in S.\ aRb\}\,. \tag{5.9}$$

5.1.2 Five Kinds of Relations

DEFINITION 5.10 **Function** A binary relation R is a function when it has the [≤ 1 arrow out] property.

$$R \text{ is a function} \Leftrightarrow \forall a \in A.\ |R(a)| \leq 1 \Leftrightarrow \forall a \in A.\ aRb \wedge aRb' \Rightarrow b = b'\,. \tag{5.10}$$

DEFINITION 5.11 **Total** A binary relation R is total when it has the [≥ 1 arrow out] property.

$$R \text{ is total} \Leftrightarrow \forall a \in A.\ |R(a)| \geq 1 \Leftrightarrow R^{-1}(B) = A\,. \tag{5.11}$$

DEFINITION 5.12 **Total Function** A binary relation R is a total function when it has the [$= 1$ arrow out] property.

$$R \text{ is a total function} \Leftrightarrow \forall a \in A.\ |R(a)| = 1\,. \tag{5.12}$$

DEFINITION 5.13 **Injective/One-to-one** A binary relation R is injective when it has the [≤ 1 arrow in] property.

$$R \text{ is injective} \Leftrightarrow \forall b \in B.\ |R^{-1}(b)| \leq 1 \Leftrightarrow \forall b \in B.\ aRb \wedge a'Rb \Rightarrow a = a'\,. \tag{5.13}$$

DEFINITION 5.14 **Surjective/Onto** A binary relation R is surjective when it has the [≥ 1 arrow in] property.

$$R \text{ is surjective} \Leftrightarrow \forall b \in B.\ |R^{-1}(b)| \geq 1 \Leftrightarrow R(A) = B\,. \tag{5.14}$$

DEFINITION 5.15 **Bijective/One-to-one correspondence** A binary relation R is bijective when it has both the [$= 1$ arrow out] and [$= 1$ arrow in] property.

$$R \text{ is bijective} \Leftrightarrow R \text{ is a function} \wedge R \text{ is total} \wedge R \text{ is injective} \wedge R \text{ is surjective}\,. \tag{5.15}$$

DEFINITION 5.16 **Permutation** A bijection from a set A to itself.

5.1.3 Functions

If a binary relation f is a function, when $(a, b) \in f$, we sometimes write $b = f(a)$, since b is uniquely determined by the choice of a, and we say that a is the **argument** of f and

b is the **value** of f at a. A function with a finite domain could be specified by a table that shows the value of the function at each element of the domain.

DEFINITION 5.17 **Partial Function** A function that there may be domain elements for which the function is not defined.

DEFINITION 5.18 **Total Function** A function is defined on every element of its domain.

DEFINITION 5.19 **Composition**

$$(g \circ f)(x) := g(f(x)) . \tag{5.16}$$

5.1.4 Isomorphic

DEFINITION 5.20 **Isomorphic** A binary relation R on a set A is isomorphic to a relation S on a set B iff there is a relation-preserving bijection from A to B; that is, there is a bijection $f \ A \mapsto B$ such that

$$\forall a, a' \in A.\ aRa' \Leftrightarrow f(a)Sf(a') . \tag{5.17}$$

5.2 Finite Cardinality

DEFINITION 5.21 **Cardinality** $|A|$ If A is a finite set, $|A| \in \mathbb{N}$ is the number of elements in A. $|\varnothing| = 0$.

DEFINITION 5.22 A **surj** B A surj $B :=$ there is a surjective function from A to B.

DEFINITION 5.23 A **inj** B A inj $B :=$ there is a injective total relation from A to B.

DEFINITION 5.24 A **bij** B A bij $B :=$ there is a bijection from A to B.

THEOREM 5.25 **Mapping Rules** For finite sets A, B

$$|A| \geq |B| \Leftrightarrow A \text{ surj } B , \tag{5.18}$$
$$|A| \leq |B| \Leftrightarrow A \text{ inj } B , \tag{5.19}$$
$$|A| = |B| \Leftrightarrow A \text{ bij } B , \tag{5.20}$$

THEOREM 5.26

$$|A| = n \Rightarrow |\operatorname{pow} A| = |\{0,1\}^n| = 2^n . \tag{5.21}$$

5.3 Infinite Sets

5.3.1 Sets Relations

DEFINITION 5.27 A **strict** B $A \text{ strict } B := \neg(A \text{ surj } B)$.

COROLLARY 5.28 For finite sets,

$$A \text{ strict } B \Leftrightarrow |A| < |B| . \tag{5.22}$$

Proof.

$$A \text{ strict } B \Leftrightarrow \neg(A \text{ surj } B) \Leftrightarrow \neg(|A| \geq |B|) \Leftrightarrow |A| < |B| . \tag{5.23}$$

□

THEOREM 5.29 Let A be a set and $c \notin A$,

$$A \text{ is infinite} \Leftrightarrow A \text{ bij } A \cup \{c\} . \tag{5.24}$$

Proof. Since A is not the same size as $A \cup \{c\}$ when A is finite, we only have to show that $A \cup \{c\}$ is the same size as A when A is infinite. That is, we have to find a bijection between $A \cup \{c\}$ and A when A is infinite.

Since A is infinite, it certainly has at least one element a_0; But since A is infinite, it certainly has at least two element, and one of them must not equal to a_0, call this new element a_1; But since A is infinite, it certainly has at least three element, and one of them must not equal to both a_0 and a_1, call this new element a_2. Continuing in this way, we conclude that there is and infinite sequence $a_0, a_1, \ldots, a_n, \ldots$ of different elements of A. Now it is easy to define a bijection $f: A \cup \{c\} \mapsto A$:

$$f(b) := a_0 , \tag{5.25}$$

$$f(a_i) := a_{i+1}, \forall i \in \mathbb{N} , \tag{5.26}$$

$$f(a) := a, \forall a \in (A - \{b, a_0, a_1, \ldots\}) . \tag{5.27}$$

□

THEOREM 5.30 For any sets A, B, C,

$$A \text{ surj } B \Leftrightarrow B \text{ inj } A , \tag{5.28}$$

$$A \text{ bij } B \Leftrightarrow B \text{ bij } A , \tag{5.29}$$

$$A \text{ surj } B \ \vee \ B \text{ surj } A , \tag{5.30}$$

$$(A \text{ surj } B) \wedge (B \text{ surj } C) \Rightarrow A \text{ surj } C , \tag{5.31}$$

$$(A \text{ bij } B) \wedge (B \text{ bij } C) \Rightarrow A \text{ bij } C , \tag{5.32}$$

$$(A \text{ strict } B) \wedge (B \text{ strict } C) \Rightarrow A \text{ strict } C , \tag{5.33}$$

$$(A \text{ surj } B) \wedge (B \text{ surj } A) \Rightarrow A \text{ bij } B\,. \quad \text{(Schröder-Bernstein)} \tag{5.34}$$

5.3.2 Countable Sets

DEFINITION 5.31 **Countable** A set A is countable iff its elements can be listed in order

$$a_0, a_1, \ldots a_n, \ldots\,. \tag{5.35}$$

Otherwise, it is **uncountable**. Finite sets are countable.

LEMMA 5.32 Let A be an infinite set, saying that A can be listed in this way is formally saying that the function $f\ \mathbb{N} \mapsto A$ defined by the rule that $\forall i \in \mathbb{N}.\, f(i) = a_i$ is a bijection:

$$A \text{ is countable infinite} \Leftrightarrow \mathbb{N} \text{ bij } A\,. \tag{5.36}$$

LEMMA 5.33 Let A be a set,

$$A \text{ is countable} \Leftrightarrow \mathbb{N} \text{ surj } A\,. \tag{5.37}$$

COROLLARY 5.34 If B is an uncountable set, and C is a countable set,

$$A \text{ surj } B \Rightarrow A \text{ is uncountable}\,. \tag{5.38}$$

$$C \text{ surj } A \Rightarrow A \text{ is countable}\,. \tag{5.39}$$

THEOREM 5.35 The following sets are countably infinite:

$$\mathbb{N}, \mathbb{N}^+, \mathbb{Z}, \mathbb{Z}^+, \mathbb{Q}, \mathbb{Q}^+, \mathbb{N} \times \mathbb{N}, \mathbb{Z} \times \mathbb{Z}, \{0,1\}^*\,. \tag{5.40}$$

THEOREM 5.36 The following sets are uncountable:

$$\mathbb{R}, \mathbb{C}\,. \tag{5.41}$$

LEMMA 5.37 Countable sets are closed under unions and Cartesian products.

LEMMA 5.38 Countable infinite sets are the "smallest" infinite sets, namely,

$$A \text{ is an infinite set} \wedge B \text{ is countable} \Rightarrow A \text{ surj } B\,. \tag{5.42}$$

LEMMA 5.39 You can add a countably infinite number of new elements to an infinite set and still wind up with just a set of the same size. Namely, if A is an infinite set, and B is a countable infinite set that has no elements in common with A, then

$$A \text{ bij } A \cup B\,. \tag{5.43}$$

5.3.3 Power Sets are Strictly Bigger

THEOREM 5.40 **Cantor's Theorem** For any set A,

$$A \text{ strict } \operatorname{pow} A\,. \tag{5.44}$$

Proof. We use proof by contradiction. Assume the theorem is false, namely there is a surjective function $f\colon A \mapsto \operatorname{pow} A$. Let

$$S := \{a \in A \mid a \notin f(a)\} \in \operatorname{pow} A\,. \tag{5.45}$$

Since f is a surjective function, then

$$\exists a_0 \in A.\, f(a_0) = S\,. \tag{5.46}$$

There are two cases.
Case 1: $a_0 \in f(a_0)$, then

$$a_0 \in f(a_0) \Rightarrow a_0 \notin S \Rightarrow a_0 \notin f(a_0)\,. \tag{5.47}$$

Case 2: $a_0 \notin f(a_0)$, then

$$a_0 \notin f(a_0) \Rightarrow a_0 \in S \Rightarrow a_0 \in f(a_0)\,. \tag{5.48}$$

That is to say,

$$a_0 \in f(a_0) \Leftrightarrow a_0 \notin f(a_0)\,, \tag{5.49}$$

which is a contradiction, namely, there is an element $S \in \operatorname{pow} A$ that is not in the range of f. □

COROLLARY 5.41 $\operatorname{pow} \mathbb{N}$ is uncountable.

THEOREM 5.42

$$\operatorname{pow} \mathbb{N} \text{ bij } \{0,1\}^{\omega}\,. \tag{5.50}$$

COROLLARY 5.43 $\{0,1\}^{\omega}$ is uncountable.

5.4 The Halting Problem

DEFINITION 5.44 **ASCII*** A finite set of strings over the 256 character ASCII alphabet.

DEFINITION 5.45 **String Procedure** A programming procedure f when it takes a string $t \in \text{ASCII}^*$ as input.

Every program f can be written as a string $s \in$ASCII*, and we denote it as f_s. You can think of f_s as the result of compiling s into something executable.

It's technically helpful to treat every string in ASCII* as a program for a string procedure. So when a string $s \in$ASCII* doesn't parse as a proper string procedure, we'll define f_s to be some default string procedure — say one that never halts on anything it is applied to.

DEFINITION 5.46 **Halting Problem** Given an arbitrary program f_s and an input t, to determine whether the program will run forever if it is not interrupted. If the program does not run forever, it is said to halt.

THEOREM 5.47 **The Halting Problem is Undecidable** There is no computational procedure for halting of arbitrary string procedures.

Proof. We use proof by contradiction. Assume the theorem is false, namely, there is a procedure h such that

$$h(s,t) = \begin{cases} \mathrm{T} & \text{if } f_s(t) \text{ halts.} \\ \mathrm{F} & \text{if } f_s(t) \text{ does not halt.} \end{cases} \tag{5.51}$$

We modify h to be h' such that

$$h'(s) = \begin{cases} \text{halts} & \text{if } h(s,s) = \mathrm{F}. \\ \text{does not halt} & \text{if } h(s,s) = \mathrm{T}. \end{cases} \tag{5.52}$$

Let s_0 to be the string for h', there are two cases:
Case 1: $h'(s_0)$ halts,

$$h'(s_0) \text{ halts} \Rightarrow h(s_0,s_0) = \mathrm{F} \Rightarrow f_{s_0}(s_0) \text{ does not halt}, \tag{5.53}$$

Case 1: $h'(s_0)$ does not halt,

$$h'(s_0) \text{ does not halt} \Rightarrow h(s_0,s_0) = \mathrm{T} \Rightarrow f_{s_0}(s_0) \text{ halts}. \tag{5.54}$$

That is to say,

$$h'(s_0) \text{ halts} \Leftrightarrow f_{s_0}(s_0) \text{ does not halt}. \tag{5.55}$$

This is a contradiction. Therefore, it is impossible to write a procedure that decides whether strings halt. Besides, there is fun video to illustrate the Halting Problem.[1] □

THEOREM 5.48 It is impossible to have a procedure that is a perfect recognizer for any overall run time property.

1 https://www.youtube.com/watch?v=92WHN-pAFCs.

For example, most compilers do static type-checking at compile time to ensure that programs will not make run-time type errors. A program that type-checks is guaranteed not to cause a run-time type-error. Since it is impossible to recognize perfectly when programs will not cause type-errors, it follows that the type-checker must be rejecting programs that really would not cause a type-error. The conclusion is that there is no perfect type-checker.

6 Number Theory

DEFINITION 6.1 **Number Theory** The study of the integers $\mathbb{Z}$.

6.1 Divisibility

DEFINITION 6.2 a **divides** b ($a \mid b$)

$$a \mid b := \exists k \in \mathbb{Z}.\ ka = b\,. \tag{6.1}$$

LEMMA 6.3

$$\forall n \in \mathbb{Z}.\ n \mid 0\,, \tag{6.2}$$
$$\forall n \in \mathbb{Z}.\ n \mid n\,, \tag{6.3}$$
$$\forall n \in \mathbb{Z}.\ \pm 1 \mid n\,, \tag{6.4}$$
$$\tag{6.5}$$

LEMMA 6.4

$$0 \mid n \Rightarrow n = 0\,. \tag{6.6}$$

LEMMA 6.5

$$a \mid b \wedge b \mid c \Rightarrow a \mid c\,, \tag{6.7}$$
$$c \mid a \wedge c \mid b \Rightarrow \forall t_1, t_2 \in \mathbb{Z}.\ c \mid t_1 a + t_2 b\,, \tag{6.8}$$
$$\forall t \neq 0 \in \mathbb{Z}.\ a \mid b \Leftrightarrow ta \mid tb\,. \tag{6.9}$$

DEFINITION 6.6 **Integer Linear Combination** An integer n is a integer linear combination of numbers $b_0, \ldots, b_k$ iff

$$\exists t_0, \ldots, t_n \in \mathbb{Z}.\ n = \sum_{i=0}^{k} t_i b_i\,. \tag{6.10}$$

THEOREM 6.7 **Division Theorem** Given $n \in \mathbb{Z}$ and $d \in \mathbb{Z} - \{0\}$, there exists a unique pair of integers q and $r \in [0, |d|)$ such that

$$n = q \cdot d + r\,, \tag{6.11}$$

where

$$q := \mathrm{qcnt}(n, d)\,, \tag{6.12}$$
$$r := \mathrm{rem}(n, d)\,. \tag{6.13}$$

6.2 The Greatest Common Divisor

DEFINITION 6.8 **Common Divisor** c is a common divisor of a and b iff $c \mid a \wedge c \mid b$.

DEFINITION 6.9 **Greatest Common Divisor (GCD)**

$$\gcd(a,b) := \max\{c \in \mathbb{Z} \mid c \mid a \wedge c \mid b\}\,. \tag{6.14}$$

LEMMA 6.10

$$\forall n \in \mathbb{Z}^+.\ \gcd(n,n) = n\,, \tag{6.15}$$
$$\forall n \in \mathbb{Z}^+.\ \gcd(n,1) = 1\,, \tag{6.16}$$
$$\forall n \in \mathbb{Z}^+.\ \gcd(n,0) = n\,. \tag{6.17}$$

6.2.1 Euclid's Algorithm

LEMMA 6.11

$$\forall b \in \mathbb{Z} - \{0\}.\ \gcd(a,b) = \gcd(b, \mathrm{rem}(a,b))\,. \tag{6.18}$$

Proof. By the Divisor Theorem,

$$a = qb + r\,. \tag{6.19}$$

So a is a linear combination of b and r, which implies that

$$c \mid b \wedge c \mid r \Rightarrow c \mid a\,. \tag{6.20}$$

Likewise, r is a linear combination of a and b, so

$$c \mid a \wedge c \mid b \Rightarrow c \mid r\,. \tag{6.21}$$

This means that a and b have the same common divisors as b and r, and so they have the same greatest common divisor. □

Euclid's Algorithm can be seen in Fig. 2. It can be modeled as a state machine.

- states: $\mathbb{N} \times \mathbb{N}$;
- start state: (a,b);
- transitions: $(x,y) \to (y, \mathrm{rem}(x,y))$.

LEMMA 6.12 Euclid's Algorithm will terminate in $\leq 2\lg a + 1$ transitions.

Algorithm 2 Euclid's Algorithm.

Input: (a,b). // wlog, assume $a \geq b > 0$
Output: $\gcd(a,b)$.

$(x,y) \leftarrow (a.b)$
while $y \neq 0$
  $(x,y) \leftarrow (y, \text{rem}(x,y))$
return x

Proof. Note that after two transitions, the first component will become $\text{rem}(x,y)$:

$$(x,y) \to (y,\text{rem}(x,y)) \to (\text{rem}(x,y),\text{rem}(y,\text{rem}(x,y)))\,. \tag{6.22}$$

We have

$$\text{rem}(x,y) \leq \frac{x}{2}\,. \tag{6.23}$$

This is because if $y \leq \frac{x}{2}$, then $\text{rem}(x,y) < y \leq \frac{x}{2}$ by definition. If $y > \frac{x}{2}$, then $\text{rem}(x,y) = x - y \leq \frac{x}{2}$. Since x starts off equal to a and gets halved or smaller every two steps, it will reach its minimum value $\gcd(a,b)$ after $\leq 2\lg a$ transitions. After that, the algorithm takes at most one more transition to terminate. In other words, Euclid's algorithm terminates after $\leq 2\lg a + 1$ transitions.

□

LEMMA 6.13 The following is a preserved invariant:

$$P((x,y)) := \gcd(x,y) = \gcd(a,b)\,, \tag{6.24}$$

THEOREM 6.14 $P((a,b))$ is a partial correctness.

Proof. So at the termination $(x,0)$, the invariant will be true and so

$$x = \gcd(x,0) = \gcd(a,b)\,. \tag{6.25}$$

□

6.2.2 Properties of the Greatest Common Divisor

LEMMA 6.15

$$\forall t_1, t_2 \in \mathbb{Z}.\ \gcd(a,b) \mid t_1 a + t_2 b\,. \tag{6.26}$$

Proof.

$$\gcd(a,b) \mid a \wedge \gcd(a,b) \mid b \Rightarrow \gcd(a,b) \mid t_1 a + t_2 b\,. \tag{6.27}$$

□

THEOREM 6.16 $\gcd(a,b)$ is the smallest linear combination of a and b,

Proof. By the Well Ordering Principle, there is a smallest positive integer linear combination of a and b. call it m, i.e.,

$$m = t_1 a + t_2 b\,. \tag{6.28}$$

We will prove that $\gcd(a,b) = m$ by showing both $\gcd(a,b) \leq m$ and $m \leq \gcd(a,b)$.

Since $\gcd(a,b)$ divides any linear combination of a and b. in particular,

$$\gcd(a,b) \mid m\,, \tag{6.29}$$

which implies that $\gcd(a,b) \leq m$.

By the Division Theorem,

$$\exists q \in \mathbb{Z}, r \in [0,m)\,.\, a = qm + r\,, \tag{6.30}$$

Substitute $m = t_1 a + t_2 b$ gives

$$a = q(t_1 a + t_2 b) + r = qt_1 a + qt_2 b + r\,, \tag{6.31}$$

so

$$r = (1 - qt_1)a - qt_2 b\,, \tag{6.32}$$

which means that $r \in [0,m)$ is a linear combination of a and b. Since m is the smallest positive linear combination of a and b, the only possibility is that $r = 0$, so

$$r = 0 \Rightarrow \text{rem}(a,m) = 0 \Rightarrow m \mid a\,. \tag{6.33}$$

By a similar argument, we can conclude that $m \mid b$, which means that r is a common divisor of a and b. Therefore, m must less than or equal to the greatest common divisor of a and b, i.e., $m \leq \gcd(a,b)$. □

When we need to proving statements involving gcd, we first translate gcd into linear combination, then argue about linear combination, and translate back to gcd at the end.

COROLLARY 6.17

$$\forall c \in \mathbb{Z}.\ (\exists t_1, t_2 \in \mathbb{Z}.\ c = t_1 a + t_2 b) \Leftrightarrow \gcd(a,b) \mid c\,. \tag{6.34}$$

COROLLARY 6.18

$$\forall k \in \mathbb{Z}^+.\ \gcd(ka,kb) = k \cdot \gcd(a,b)\,. \tag{6.35}$$

Algorithm 3 The Pulverizer.

Input: (a, b). // wlog, assume $a \geq b > 0$
Output: (s_1, s_2) such that $\gcd(a, b) = s_1 a + s_2 b$.

$(x, y) \leftarrow (a.b)$
$(s_1, s_2) \leftarrow (1, 0)$ // $x = s_1 a + s_2 b$
$(t_1, t_2) \leftarrow (0, 1)$ // $y = t_1 a + t_2 b$
while $y \neq 0$
  $q \leftarrow \text{qcnt}(x, y)$
  $(x, y) \leftarrow (y, \text{rem}(x, y))$
  $(s_1, s_2) \leftarrow (t_1, t_2)$
  $(t_1, t_2) \leftarrow (s_1 - t_1 q, s_2 - t_2 q)$
return (s_1, s_2)

COROLLARY 6.19

$$c \mid a \wedge c \mid b \Leftrightarrow c \mid \gcd(a, b)\,. \tag{6.36}$$

COROLLARY 6.20

$$\gcd(a, b) = 1 \wedge \gcd(a, c) = 1 \Rightarrow \gcd(a, bc) = 1\,. \tag{6.37}$$

Proof.

$$\exists t_1, t_2 \in \mathbb{Z}.\ t_1 a + t_2 b = 1\,, \tag{6.38}$$
$$\exists s_1, s_2 \in \mathbb{Z}.\ s_1 a + s_2 c = 1\,. \tag{6.39}$$

Multiplying these two gives

$$(t_1 a + t_2 b)(s_1 a + s_2 c) = (a t_1 s_1 + b t_2 s_1 + c t_1 s_2) a + (t_2 s_2)(bc) = 1\,. \tag{6.40}$$

This is a linear combination of a and bc that is equal to 1, so $\gcd(a, bc) = 1$. □

COROLLARY 6.21

$$a \mid bc \wedge \gcd(a, b) = 1 \Rightarrow a \mid c\,. \tag{6.41}$$

6.2.3 The Pulverizer

$\gcd(a, b)$ can be expressed as a linear combination of a and b, the Pulverizer is used to find the linear combination coefficients of a and b. The Pulverizer goes through the same steps of Euclid's Algorithm, but as we compute $\gcd(x, y)$, we keep track of how to write each remainder $\text{rem}(x, y)$ as a linear combination of a and b. Our objective is to write the

last nonzero remainder, which is the $\gcd(a,b)$, as such a linear combination. See Alg. 3. It can be modeled as a state machine.

- states: $\mathbb{N}^6$;
- start state: $(a,b,1,0,0,1)$;
- transitions:

$$(x,y,s_1,s_2,t_1,t_2) \to (y,\text{rem}(x,y),t_1,t_2,s_1-t_1q,s_2-t_2q)\,, \tag{6.42}$$

where $q = \text{qcnt}(x,y)$.

LEMMA 6.22 The Pulverizer will termination in $O(\lg a)$ steps.

Proof. Since the Pulverizer requires only a little more computation than Euclid's algorithm, you can "pulverize" very large numbers very quickly by using this algorithm. □

LEMMA 6.23 The following is a preserved invariant

$$P((x,y,s_1,s_2,t_1,t_2)) := \gcd(x,y) = \gcd(a,b) \wedge x = s_1a + s_2b \wedge y = t_1a + t_2b\,. \tag{6.43}$$

THEOREM 6.24 $P((x,y,s_1,s_2,t_1,t_2))$ is a partial correctness.

Proof. Since $P((x,y,s_1,s_2,t_1,t_2))$ is a preserved invariant, and $P((a,b,1,0,0,1))$ is true. So at the termination $(x,0,s_1,s_2,t_1,t_2)$, the invariant will be true and so

$$x = \gcd(x,0) = \gcd(a,b) \wedge x = s_1a + s_2b\,. \tag{6.44}$$

□

Using The Pulverizer, we can write $\gcd(a,b)$ as a linear combination of a and b:

$$\gcd(a,b) = s_1a + s_2b\,. \tag{6.45}$$

The coefficient s_1 could be either positive or negative. However, this is equivalent to the linear combination

$$\gcd(a,b) = (s_1 + kb)a + (s_2 - ka)b\,. \tag{6.46}$$

By adjusting k, we can get $s_1 + kb > 0$.

6.3 Primes

6.3.1 Primes and Composites

DEFINITION 6.25 **Prime** A number greater than 1 that is divisible only by itself and 1.

Algorithm 4 Primality Testing (Naive).

input: n. // assume $n \geq 2$
output: whether n is prime.

```
for i ← 2 to ⌊√n⌋
  if i | n
    return false
return true
```

DEFINITION 6.26 **Composite** A number other than 0, 1, and -1 that is not a prime. So 0, 1, and -1 are the only integers that are neither prime nor composite.

DEFINITION 6.27 **Primality Testing** Determine whether a large number n is prime or composite.

A naive way for primality testing can be seen in Alg.̃refPrimality Testing (Naive).. It takes $O(\sqrt{n}) = O\left(\sqrt{2}^{\lg n}\right)$ time, which is exponential in the size of n measured by the number of digits in the binary representation of n. There is a simple, fast probabilistic primality test (see the Fermat's Little Theorem part).

DEFINITION 6.28 **Relatively Prime** a and b are relatively prime iff $\gcd(a, b) = 1$.

6.3.2 Prime Number Theorem

DEFINITION 6.29 $\pi(n)$ The number of primes up to n.

$$\pi(n) := \sum_{k=2}^{n} \mathbb{I}\{k \text{ is prime}\}. \tag{6.47}$$

THEOREM 6.30 **Prime Number Theorem** The overall growth of rate of $\pi(n)$ is the same as $\frac{n}{\log n}$. Thus, primes gradually taper off.

$$\pi(n) = \Theta\left(\frac{n}{\log n}\right). \tag{6.48}$$

As a rule of thumb, about 1 integer out of every $\log n$ integers in the vicinity of n is a prime.

THEOREM 6.31 **Chebyshev's Theorem on Prime Density**

$$\forall n > 1 \in \mathbb{Z}.\ \pi(n) > \frac{n}{3 \log n}. \tag{6.49}$$

6.3.3 Conjectured Inefficiency of Factoring

Given the product of two large primes $n = pq$, there is no efficient procedure to recover the primes p and q. That is, no polynomial time procedure is guaranteed to find p and q in a number of steps bounded by a polynomial in the length of the binary representation of n (not n itself). The length of the binary representation at most $1 + \lg n$. The best algorithm known is the *number field sieve*, which runs in $O\left(\exp 1.9(\log n)^{\frac{1}{3}}(\log\log n)^{\frac{2}{3}}\right)$ time. This algorithm is infeasible when n has 300 digits or more.

We can factor any number n into the product of two prime numbers using a SAT solver. We can build two digital circuits

- A multiplier circuit: Two k bits inputs x and y and one $2k$ bits output. It compute the product of two input, namely xy.
- A equality circuit: One $2k$ bits input and one 1 bit output. It tests whether the input is n.

Then we can make these two circuits in cascade.

The procedure is as follows. Set the first bit of x to be 1. Do a SAT test to see if there is a satisfying assignment of values for the remaining $2k-1$ inputs used for the x and y representations to cause the circuit to give output 1. If there is such an assignment, fix the first bit of x to be 1, otherwise fix it to be 0. Now do the same thing to fix the second bit of x, and then third, proceeding in this way through all the k inputs for x. At this point, we have the complete k-bit binary representation of x that is a factor of n. We can now find y by dividing n by x. So after k SAT tests, we have factored n. This means that if SAT could be solved in polynomial time, then so could factoring.

6.3.4 Fundamental Theorem of Arithmetic

LEMMA 6.32 Suppose p is prime,

$$p \mid ab \Rightarrow p \mid a \vee p \mid b\,. \tag{6.50}$$

Proof. There are two cases:
Case 1: $\gcd(a,p) = p$. Then the claim holds since $p \mid a$.
Case 2: $\gcd(a,p) \neq p$. In this case $\gcd(a,p) = 1$ since 1 and p are the only positive divisors of p. Since $\gcd(a,p)$ is a linear combination of a and p, so

$$\exists t_1, t_2 \in \mathbb{Z}.\ 1 = t_1 a + t_2 p\,. \tag{6.51}$$

Multiplying b on both sides,

$$b = t_1 ab + t_2 bp = t_1(ab) + (t_2 b)p\,, \tag{6.52}$$

that is, b is a linear combination of ab and p. Then

$$p \mid ab \wedge p \mid p \Rightarrow p \mid b\,. \tag{6.53}$$

□

LEMMA 6.33 Suppose p is prime,

$$p \mid a_1 a_2 \cdots a_n \Rightarrow \exists i.\, p \mid a_i\,. \tag{6.54}$$

Proof. By induction on n. □

DEFINITION 6.34 **Weakly Decreasing Sequence of Numbers** A sequence of numbers with each number in the sequence is at least as big as the numbers after it. A sequence of just one number as well as a sequence of no numbers, i.e. the empty sequence, is weakly decreasing by this definition.

THEOREM 6.35 **Fundamental Theorem of Arithmetic/Unique Factorization Theorem** Each positive integer greater than 1 can factors uniquely into a weakly decreasing sequence of primes.

$$\forall n > 1 \in \mathbb{Z}, \exists p_1, p_2, \ldots, p_k \in \text{Primes}.\ n = p_1 p_2 \cdots p_k \wedge p_1 \geq p_2 \geq \cdots \geq p_k\,. \tag{6.55}$$

Proof. We have already showed that every positive integer can be expressed as a product of primes. So we just have to prove this expression is unique.

We use proof by contradiction. Assume the claim is false. By the Well Ordering Principle, there exists a smallest positive integer n such that it can be written as products of primes in more than one way:

$$n = \prod_{i=1}^{m} p_i = \prod_{j=1}^{n} q_j\,, \tag{6.56}$$

where both products are in weakly decreasing order and $p_1 \leq q_1$. There are two cases.
Case 1: $p_1 = q_1$, then $\frac{n}{p_1}$ would also be the product of different weakly decreasing sequences of primes, namely,

$$\frac{n}{p_1} = \prod_{i=2}^{m} p_i = \prod_{j=2}^{n} q_j\,. \tag{6.57}$$

Since $\frac{n}{p_1} < n$, this is a contradiction.
Case 1: $p_1 < q_1$. Since p_i's are weakly decreasing, all the p_i's are less than q_1. Since $q_1 \mid n$, then

$$q_1 \mid \prod_{i=1}^{m} p_i \Rightarrow \exists i.\, q_1 \mid p_i\,, \tag{6.58}$$

which contradicts the fact that q_1 is bigger than all them. □

6.4 Modular Arithmetic

6.4.1 Congruence

DEFINITION 6.36 **Congruence** ($a \equiv b \pmod{n}$)

$$a \equiv b \pmod{n} := n \mid a - b\,. \tag{6.59}$$

LEMMA 6.37 [Remainder Lemma]

$$a \equiv b \pmod{n} \Leftrightarrow \mathrm{rem}(a,n) = \mathrm{rem}(b,n)\,. \tag{6.60}$$

Proof. By the Division Theorem

$$\exists q_1 \in \mathbb{Z}, r_1 \in [0,n)\,.\, a = q_1 n + r_1\,, \tag{6.61}$$

$$\exists q_2 \in \mathbb{Z}, r_2 \in [0,n)\,.\, b = q_2 n + r_2\,. \tag{6.62}$$

Subtracting these two gives,

$$a - b = (q_1 - q_2)n + (r_1 - r_2)\,, \tag{6.63}$$

where $r_1 - r_2 \in (-n, n)$.

$$a \equiv b \pmod{n} \Leftrightarrow n \mid a - b \Leftrightarrow n \mid (q_1 - q_2)n + (r_1 - r_2) \Leftrightarrow n \mid r_1 - r_2\,. \tag{6.64}$$

Therefore, $r_1 - r_2$ must in fact equal to 0, namely,

$$\mathrm{rem}(a,n) = \mathrm{rem}(b,n)\,. \tag{6.65}$$

□

LEMMA 6.38 Congruence is an equivalence relation.

$$a \equiv a \pmod{n}\,, \quad \text{(reflexivity)} \tag{6.66}$$

$$a \equiv b \Leftrightarrow b \equiv a \pmod{n}\,, \quad \text{(symmetry)} \tag{6.67}$$

$$a \equiv b \wedge b \equiv c \Rightarrow a \equiv c \pmod{n}\,. \quad \text{(transitivity)} \tag{6.68}$$

LEMMA 6.39 [Congruence Lemma] Congruences are preserved by addition and multiplication.

$$a \equiv b \wedge c \equiv d \pmod{n} \Rightarrow a + c \equiv b + d \pmod{n}\,, \tag{6.69}$$

$$a \equiv b \wedge c \equiv d \pmod{n} \Rightarrow ac \equiv bd \pmod{n}\,. \tag{6.70}$$

Proof.

$$a \equiv b \pmod{n} \Rightarrow n \mid b-a \Rightarrow n \mid (b+c)-(a+c) \Rightarrow a+c \equiv b+c \pmod{n} \tag{6.71}$$

The same reasoning leads to

$$c \equiv d \pmod{n} \Rightarrow b+c \equiv b+d \pmod{n}. \tag{6.72}$$

Now transitivity gives

$$a+c \equiv b+c \equiv b+d \pmod{n}. \tag{6.73}$$

□

6.4.2 Remainder Arithmetic

LEMMA 6.40

$$a \equiv \text{rem}(a,n) \pmod{n}. \tag{6.74}$$

Congruence modulo n defines a partition of the integers into n sets so that congruent numbers are all in the same set. When arithmetic is done modulo n, there are really only n different kinds of numbers to worry about, because there are only n possible remainders in $[0,n)$.

LEMMA 6.41

$$\text{rem}(a+b,n) = \text{rem}(\text{rem}(a,n)+\text{rem}(b,n),n), \tag{6.75}$$

$$\text{rem}(a\cdot b,n) = \text{rem}(\text{rem}(a,n)\cdot\text{rem}(b,n),n). \tag{6.76}$$

Proof.

$$a \equiv \text{rem}(a,n) \pmod{n} \wedge b \equiv \text{rem}(b,n) \pmod{n} \Rightarrow a+b \equiv \text{rem}(a,n)+\text{rem}(b,n) \pmod{n}, \tag{6.77}$$

and the remainders on each side of this congruence are equal. □

THEOREM 6.42 **General Principle of Remainder Arithmetic** To find the remainder on division by n of the result of a series of additions and multiplications, applied to some integers

- Replace each integer operand by its remainder on division by n.
- Keep each result of an addition or multiplication in the range $[0,n)$ by immediately replacing any result outside that range by its remainder on division by n.

DEFINITION 6.43 $+ \quad (\mathbb{Z}_n)$

$$a + b \quad (\mathbb{Z}_n) := \text{rem}(a + b, n) . \tag{6.78}$$

DEFINITION 6.44 $\cdot \quad (\mathbb{Z}_n)$

$$a \cdot b \quad (\mathbb{Z}_n) := \text{rem}(a \cdot b, n) . \tag{6.79}$$

DEFINITION 6.45 **The Ring of Integers Modulo n $(\mathbb{Z}_n)$** The set of integers in the range $[0, n)$ together with the operations $+ \quad (\mathbb{Z}_n)$ and $\cdot \quad (\mathbb{Z}_n)$ is referred to as $\mathbb{Z}_n$.

THEOREM 6.46 **Rules for $\mathbb{Z}_n$** $\mathbb{Z}_n$ is a commutative ring.

$$(a + b) + c = a + (b + c) \quad (\mathbb{Z}_n) , \quad \text{(associativity)} \tag{6.80}$$

$$a + b = b + a \quad (\mathbb{Z}_n) , \quad \text{(commutativity)} \tag{6.81}$$

$$0 + a = a \quad (\mathbb{Z}_n) , \quad \text{(identity)} \tag{6.82}$$

$$(a \cdot b) \cdot c = a \cdot (b \cdot c) \quad (\mathbb{Z}_n) , \quad \text{(associativity)} \tag{6.83}$$

$$a \cdot b = b \cdot a \quad (\mathbb{Z}_n) , \quad \text{(commutativity)} \tag{6.84}$$

$$1 \cdot a = a \quad (\mathbb{Z}_n) , \quad \text{(identity)} \tag{6.85}$$

$$a \cdot (b + c) = a \cdot b + a \cdot c_n \quad (\mathbb{Z}_n) . \quad \text{(distributivity)} \tag{6.86}$$

DEFINITION 6.47 $\mathbb{Z}_n^*$

$$\mathbb{Z}_n^* := \{k \in \mathbb{Z}_n \mid \gcd(k, n) = 1\} . \tag{6.87}$$

COROLLARY 6.48 Suppose p is prime,

$$\mathbb{Z}_p^* = \mathbb{Z}_p - \{0\} . \tag{6.88}$$

6.4.3 Multiplicative Inverses and Canceling

DEFINITION 6.49 **Multiplicative Inverses** The multiplicative inverse of k in $\mathbb{Z}_n$ is the number k^{-1}, if exists, such that

$$k k^{-1} = 1 \quad (\mathbb{Z}_n) . \tag{6.89}$$

DEFINITION 6.50 **Cancellable** A number k is cancellable iff

$$ka = kb \quad (\mathbb{Z}_n) \Rightarrow a = b \quad (\mathbb{Z}_n) . \tag{6.90}$$

THEOREM 6.51 The following are equivalent for $k \in \mathbb{Z}_n$

- $k \in \mathbb{Z}_n^*$.
- k has an inverse in $\mathbb{Z}_n$, and the inverse k^{-1} is unique.
- k is cancellable in $\mathbb{Z}_n$.

Algorithm 5 Multiplicative Inverse.

Input: $k \in \mathbb{Z}_n^*$, n.
Output: $k^{-1} \in \mathbb{Z}_n$.

$(t_1, t_2) \leftarrow \text{pulverize}(n, k)$
$k^{-1} \leftarrow \text{rem}(t_1, n)$
return k^{-1}

Proof. **Part 1**:

$$k \in \mathbb{Z}_n^* \Rightarrow \gcd(k,n) = 1 \Rightarrow \exists t_1, t_2 \in \mathbb{Z}.\ t_1 n + t_2 k = 1\,. \tag{6.91}$$

Applying the General Principle of Remainder Arithmetic, we get

$$\text{rem}(t_1, n)\,\text{rem}(n, n) + \text{rem}(t_2, n)\,\text{rem}(k, n) = 1 \quad (\mathbb{Z}_n)\,. \tag{6.92}$$

Since $\text{rem}(n,n) = 0$ and $\text{rem}(k,n) = k$, we get

$$\text{rem}(t_2, n)k = 1 \quad (\mathbb{Z}_n) \tag{6.93}$$

Thus $\text{rem}(t_2, n)$ is the k^{-1}, and k^{-1} can be computed by The Pulverizer, see Alg. 5.
Part 2: We use proof by contradiction. Assume the claim is false, namely, suppose k_1 and k_2 are both inverses of k in $\mathbb{Z}_n$, then

$$k_1 = k_1 \cdot 1 = k_1 \cdot (k \cdot k_2) = (k_1 \cdot k) \cdot k_2 = 1 \cdot k_2 = k_2 \quad (\mathbb{Z}_n)\,, \tag{6.94}$$

which is a contradiction, therefore, the inverse is unique.
Part 3: By multiplying its inverse. □

When p is prime, each nonzero $k \in \mathbb{Z}_p$ is relatively prime to p. Therefore, each k has an inverse.

6.4.4 Euler's Theorem

DEFINITION 6.52 **Euler's Totient Function** For $n \in \mathbb{Z}^+$,

$$\phi(n) := |\mathbb{Z}_n^*| = \sum_{k=0}^{n-1} \mathbb{I}\{\gcd(k,n) = 1\}\,. \tag{6.95}$$

COROLLARY 6.53 If p is prime,

$$\phi(p) = |\mathbb{Z}_p^*| = p - 1\,, \tag{6.96}$$

since every positive number in $[0, p)$ is relatively prime to p.

Algorithm 6 Primality Test (Probabilistic).

input: n. // assume $n \geq 2$
output: whether n is prime.

for $t \leftarrow 1$ **to** T // try T times
  Randomly pick a number $k \in \mathbb{Z}_n - \{0\}$.
  if $k^{n-1} \neq 1 \quad (\mathbb{Z}_n)$
    return false
return true

COROLLARY 6.54 If p is prime,

$$\forall k \geq 1 \in \mathbb{Z}.\ \phi(p^k) = p^k - p^{k-1}\,. \tag{6.97}$$

LEMMA 6.55 [ϕ is Multiplicative]

$$\gcd(a,b) = 1 \Rightarrow \phi(ab) = \phi(a)\phi(b)\,. \tag{6.98}$$

THEOREM 6.56 **Euler's Theorem**

$$\forall k \in \mathbb{Z}_n^*.\ k^{\phi(n)} = 1 \quad (\mathbb{Z}_n)\,. \tag{6.99}$$

COROLLARY 6.57

$$\forall k \in \mathbb{Z}_n^*.\ k^{-1} = k^{\phi(n)-1} \quad (\mathbb{Z}_n)\,. \tag{6.100}$$

We can use fast exponentiation to compute $k^{\phi(n)-1}$, and compute $\phi(n)$ is easy once we know the prime factorization of n. But we know that finding the factors of n is generally hard to do when n is large, and so The Pulverizer remains the best approach to computing inverses modulo n.

THEOREM 6.58 **Fermat's Little Theorem** Suppose p is prime,

$$\forall k \in \mathbb{Z}_p^*.\ k^{p-1} = 1 \quad (\mathbb{Z}_p)\,. \tag{6.101}$$

We can use Fermat's Little Theorem as a probabilistic primality test, see Alg. 6.

- If $k^{n-1} \neq 1 \quad (\mathbb{Z}_n)$, then n is not a prime.
- If $k^{n-1} = 1 \quad (\mathbb{Z}_r)$, it might be that I just hit an n that happened to satisfy Fermat's equation even though n was not prime.

But if n is not prime, then half of the numbers in $\mathbb{Z}_n - \{0\}$ are not going to pass the Fermat test. Therefore, for a random number $k \in \mathbb{Z}_n - \{0\}$,

$$\Pr\{n \text{ is not prime} \wedge k^{n-1} = 1 \quad (\mathbb{Z}_n)\} \leq \frac{1}{2}\,. \tag{6.102}$$

Algorithm 7 Turing's Code (Version 1).

input: message m, which is a prime.

Beforehand

The sender and receiver agree on a secret key, which is a large prime k.

Encryption

$\tilde{m} \leftarrow mk$ // sender encrypts the message m

Decryption

$m \leftarrow \tilde{m}/k$ // receiver decrypts $\tilde{m}$

So I try it 50 times,

$$\Pr\{n \text{ is not prime} \wedge \forall i \in \mathbb{Z}_{50}.\, k_i^{n-1} = 1 \quad (\mathbb{Z}_n)\} \leq \frac{1}{2^{50}}\,. \tag{6.103}$$

6.5 Encryption

Encryption makes use of one-way functions that are easy to compute but hard to invert. In particular, factoring is hard.

The first step is to translate a text message into an integer so that we can perform mathematical operations on it. And we want m to be a prime number, so we may need to pad the result with some digits to make a prime $m \in \mathbb{Z}$.

6.5.1 Turing's Code (Version 1)

See Alg. 7.

Is Turing's Code safe? The Nazis see only the encrypted message $\tilde{m}$, so recovering the original message m requires factoring $\tilde{m}$, and no really efficient factoring algorithm has ever been found.

Breaking Turing's Code. Suppose Nazis has two encrypted messages sent using the same key

$$\tilde{m}_1 = m_1 k\,, \tag{6.104}$$

$$\tilde{m}_2 = m_2 k\,. \tag{6.105}$$

Then

$$\gcd(\tilde{m}_1, \tilde{m}_2) = k\,. \tag{6.106}$$

and gcd of two numbers can be computed very efficiently. So after the second message is sent, the Nazis can recover the secret key and read every message!

Algorithm 8 Turing's Code (Version 2).

input: message $m \in \mathbb{Z}_p$, which is a prime.

Beforehand

The sender and receiver agree on a large prime number p, which may be made public. As in Version 1, they also agree that some prime number $k \in \mathbb{Z}_p$ will be the secret key.

Encryption

$\tilde{m} \leftarrow mk \quad (\mathbb{Z}_p)$ // sender encrypts the message m

Decryption

$m \leftarrow \tilde{m}k^{-1} \quad (\mathbb{Z}_p) \quad // \; \tilde{m}k^{-1} = mkk^{-1} = m \quad (\mathbb{Z}_p)$

6.5.2 Turing's Code (Version 2)

See Alg. 8.

Breaking Turing's Code (known-plaintext attack). Once we known both m and $\tilde{m}$, we can get k by computing

$$m^{-1}\tilde{m} = m^{-1}mk = k \quad (\mathbb{Z}_p)\,. \tag{6.107}$$

6.5.3 RSA

DEFINITION 6.59 **Public Key Cryptography System** The sender and receiver of an encrypted message need not meet beforehand to agree on a secret key. Rather, the receiver has both a private key, which they keep secret, and a public key, which they make publicly available. A sender wishing to transmit a secret message to the receiver encrypts their message using the receiver's public key. The receiver can then decrypt the received message using their closely held private key.

See Alg 9.

Proof. Because

$$d = e^{-1} \quad (\mathbb{Z}_{\phi(n)})\,, \tag{6.108}$$

so

$$ed = 1 \quad (\mathbb{Z}_{\phi(n)})\,. \tag{6.109}$$

That means

$$\exists t \in \mathbb{Z}.\; ed = 1 + t\phi(n)\,. \tag{6.110}$$

Therefore

$$\tilde{m}^d = m^{ed} = m^{1+t\phi(n)} = m \cdot (m^{\phi(n)})^t = m \cdot 1^t = m \quad (\mathbb{Z}_n)\,. \tag{6.111}$$

Algorithm 9 RSA.

input: message $m \in \mathbb{Z}_n^*$.

Beforehand

The receiver generate two distinct primes p and q, and they must be kept hidden.

$n \leftarrow pq$.

Select an integer $e \in \mathbb{Z}_{\phi(n)}^*$ // the public key is the pair (e, n)

$d \leftarrow e^{-1} \quad (\mathbb{Z}_{\phi(n)})$ // the private key

Encryption

$\tilde{m} \leftarrow m^e \quad (\mathbb{Z}_n)$ // sender encrypts the message m

Decryption

$m \leftarrow \tilde{m}^d \quad (\mathbb{Z}_n)$ // receiver decrypts $\tilde{m}$

□

Receiver's abilities

- Find two large primes p and q: Okay because there are lots of primes and we have fast primality test.
- Compute $\phi(n)$: Okay because $\phi(n) = \phi(pq) = (p-1)(q-1)$.
- Select an integer $e \in \mathbb{Z}_{\phi(n)}^*$: Okay because there are lots of relative prime numbers and gcd is easy to compute.
- Compute $d = e^{-1} \quad (\mathbb{Z}_{\phi(n)})$: Okay by The Pulverizer.

Is RSA safe? Factor n into pq is hard with hundreds of digits.

II GRAPHS

7 Directed Graphs

7.1 Digraph Basis

DEFINITION 7.1 **Directed Graph/Digraph** A directed graph/digraph G is a pair (V, E), where $V \neq \varnothing$ is a finite set called **vertex set**, and E is a binary relation on V called **edge set**. An element of V is called a **vertex/node**. An element of E is called a **directed edge/edge/arrow**. A directed edge starts at some vertex u called the tail of the edge, and ends at some vertex v called the head of the edge. Such an edge can be represented by the ordered pair (u, v). The notation $u \to v$ denotes this edge.

DEFINITION 7.2 **Simple Digraph** Directed graph without any loops and multiple edges.

DEFINITION 7.3 **Adjacency Matrix** If a graph G has n vertices $v_1, \ldots, v_n$, we can represent the graph by its adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, where

$$A_{ij} = \mathbb{I}\{(v_i, v_j) \in E\}, \forall i, j\,. \tag{7.1}$$

DEFINITION 7.4 **In-degree** The in-degree of a vertex in a digraph is the number of arrows coming into it.

$$\operatorname{indeg} v = \sum_{e \in E} \mathbb{I}\{\operatorname{head} e = v\}\,. \tag{7.2}$$

DEFINITION 7.5 **Sink** Vertex with outdegree 0.

DEFINITION 7.6 **Out-degree** The out-degree of a vertex in a digraph is the number of arrows coming out of it.

$$\operatorname{outdeg} v = \sum_{e \in E} \mathbb{I}\{\operatorname{tail} e = v\}\,. \tag{7.3}$$

DEFINITION 7.7 **Source** Vertex with indegree 0.

LEMMA 7.8

$$\sum_{v \in V} \operatorname{indeg} v = \sum_{v \in V} \operatorname{outdeg} v = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} = |E|\,. \tag{7.4}$$

7.2 Walks and Paths

7.2.1 Walks and Paths

DEFINITION 7.9 **Walk** A walk from x to y in a digraph G is a sequence of vertices

$$(v_0, v_1, \ldots, v_k)\,, \tag{7.5}$$

such that $v_0 = x$, $v_k = y$, and

$$\forall i \in [0, k-1] \,.\, (v_i, v_{i+1}) \in E \,. \tag{7.6}$$

The walk is said to start at x, to end at y, and the **length** of the walk k is defined to be the number be edges in the path. There is always a 0-length walk from x to x.

DEFINITION 7.10 **Path** The walk is a path iff all the v_i's are different, that is,

$$i \neq j \Rightarrow v_i \neq v_j \,. \tag{7.7}$$

DEFINITION 7.11 **Closed Walk** A closed walk is a walk that begins and ends at the same vertex.

DEFINITION 7.12 **Cycle** A cycle is a positive length closed walk whose vertices are distinct except for the beginning and end vertices.

DEFINITION 7.13 **Self-loop** When a node has an edge leading back to itself, it is a length one cycle.

A single vertex counts as a length zero walk/path that begins and ends at itself. It also is a closed walk, but does not count as a cycle.

DEFINITION 7.14 **Strongly Connected** A digraph is strongly connected iff there is a path between every pair of distinct vertices.

7.2.2 Walk Relations

DEFINITION 7.15 **Reachable** When there is a walk p from vertex u to vertex v, we say that v is reachable from u, or equivalently, that u is connected to v, which we sometimes write as $x \overset{p}{\rightsquigarrow} y$.

DEFINITION 7.16 **Subwalk** A subwalk of a walk $p = (v_0, v_1, \ldots, v_k)$ is a contiguous subsequence of its vertices. That is, for any $0 \leq i \leq j \leq k$, the subsequence of vertices $(v_i, v_{i+1}, \ldots, v_j)$ is a subwalk of p.

DEFINITION 7.17 **Walk Relation G^*** For any digraph G, the walk relation G on V is defined where uG^*v iff there is a walk in G from u to v.

DEFINITION 7.18 **Positive Walk Relation G^+** uG^+v iff there is a positive length walk in G from u to v.

DEFINITION 7.19 **Length-k Walk Relation G^k** uG^kv iff there is a length-k walk in G from u to v.

If we let G^k denotes the composition of G with itself k times, then G^k is the length-k walk relation. Therefore

$$G^* = G^0 \cup G^1 \cup \cdots \cup G^{n-1}. \tag{7.8}$$

7.2.3 Walk Counting Matrix

DEFINITION 7.20 **Walk Counting Matrix** If a graph G has n vertices $v_1, \ldots, v_n$, the length-k walk counting matrix $\boldsymbol{C} \in \mathbb{R}^{n \times n}$ satisfies

$$C_{ij} = \text{The number of length-}k\text{ walks from } v_i \text{ to } v_j, \forall i,j. \tag{7.9}$$

LEMMA 7.21 If $\boldsymbol{C}$ is the length-k walk counting matrix for a graph G, and $\boldsymbol{D}$ is the length-l walk counting matrix, then $\boldsymbol{CD}$ is the length-$(k+l)$ walk counting matrix for G.

Proof. Any length-$(k+l)$ walk between vertices x and y begins with a length-k walk starting at x and ending at some vertex v, followed by a length-l walk starting at v and end at y.

So the number of length-$(k+l)$ walks from x to y that go through v at the k-th step equals the number C_{xv} of length-k walks from x to v, times the number D_{vy} of length-l walks from v to y. Then summing over all possible v gives the number of length-$(k+l)$ walks from x to y:

$$\sum_v C_{xv} D_{vy}, \tag{7.10}$$

□

THEOREM 7.22 The length-k counting matrix of a digraph G is $\boldsymbol{A}^k, \forall k \in \mathbb{N}$. In particular, $\boldsymbol{A}^0 = \boldsymbol{I}$ is the length-0 walk counting matrix.

Proof. By induction on k. □

7.2.4 Finding a (Shortest) Path

THEOREM 7.23 The shortest walk from one vertex to another is a (shortest) path.

Proof. If there is a walk from vertex x to another vertex y, and $x \neq y$, then by Well Ordering Principle, there must be a minimum length walk p from x to y. We claim that p is a path.

We use proof by contradiction. Assume the claim is false, namely, there is some vertex v occurs twice on this walk. Thai is

$$x \rightsquigarrow v \rightsquigarrow v \rightsquigarrow y. \tag{7.11}$$

Deleting $v \rightsquigarrow v$ yields a strictly shorter walk

$$x \rightsquigarrow v \rightsquigarrow y \tag{7.12}$$

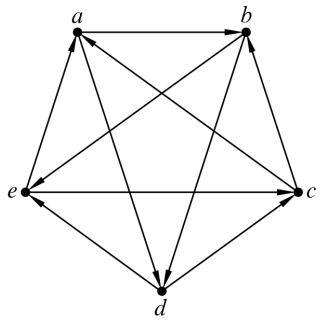

Figure 7.1
A 5-node tournament digraph.

from x to y, contradicting the minimality of p. □

THEOREM 7.24 The shortest positive length closed walk through a vertex is a cycle through that vertex.

DEFINITION 7.25 **Distance** The distance $\text{dist}(u, v)$ in a graph from vertex u to vertex v is the length of a shortest path from u to v.

THEOREM 7.26 If there is a shortest path from a_i to a_j, then

$$\text{dist}(a_i, a_j) = \min\{k \in \mathbb{Z}_n \mid A_{ij}^k > 0\}. \tag{7.13}$$

THEOREM 7.27 **The Triangle Inequality**

$$\forall u, v, w \in V.\ \text{dist}(u, v) \leq \text{dist}(u, w) + \text{dist}(w, v). \tag{7.14}$$

The equality holds true exactly when w is on a shortest path from u to v.

Proof. The length of the walk $u \rightsquigarrow w \rightsquigarrow v$ is $\text{dist}(u, w) + \text{dist}(w, v)$. This sum is an upper bound on the length of the shortest path from u to v. □

7.3 Tournament Digraph

DEFINITION 7.28 **Tournament Digraph** A tournament digraph is a digraph where for every pair of distinct vertices u and v, there is either an edge $u \to v$ xor an edge $v \to u$ (not both). See Fig.7.1 for an example.

Suppose that n players compete in a round-robin tournament and that for every pair of players u and v, either u beats v ($u \to v$) or v beats u ($v \to u$). It can be represented as a tournament digraph.

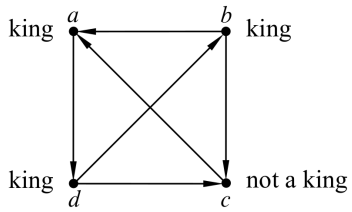

Figure 7.2
A 4-chicken tournament in which chickens a, b, and d are kings.

DEFINITION 7.29 **Directed Hamiltonian Path** A path that visit every vertex exactly once.

THEOREM 7.30 Every tournament graph contains a directed Hamiltonian path.

Proof. By strong induction on number n of vertices. Let the inductive hypothesis be $P(n) :=$ every tournament graph with n vertices contains a directed Hamiltonian path.
Base case: $P(1)$ is true since every graph with a single vertex has a Hamiltonian path consisting of only that vertex.
Inductive step: We assume that $P(1), P(2), \ldots, P(n)$ are all true. Consider a tournament graph $G = (V, E)$ with $n + 1$ vertices. Select one vertex v arbitrarily. Every other vertex in the tournament graph either has an edge to v or from v. Thus, we can partition the remaining vertices into two corresponding sets $V_T := \{u \mid (u, v) \in E\}$ and $V_F := \{u \mid (v, u) \in E\}$.

The vertices in V_T together with the edges that join from them form a smaller tournament graph. Thus, by strong induction, there is a Hamiltonian path within V_T. Similarly, there is a Hamiltonian path within V_F. Joining the path in V_T to the vertex v followed by the path in V_F gives a Hamiltonian path through the whole tournament. As special cases, if V_T of V_F is empty, then so is the corresponding portion of the path. □

DEFINITION 7.31 **Directed Eulerian Path** A path that visit every edge exactly once.

DEFINITION 7.32 **The King Chicken Problem** Suppose that there are n chickens which formed a tournament graph. That is to say, for each pair of distinct chickens u and v, either u pecks v ($u \to v$) or v pecks u ($v \to u$), but not both. We say that chicken u virtually pecks chicken v ($u \rightsquigarrow v$) if either:

- $u \to v$
- $\exists w.\ u \rightsquigarrow w \wedge w \rightsquigarrow v$.

A chicken that virtually pecks every other chicken is called a king chicken. See Fig. 7.2 for an example.

THEOREM 7.33 **King Chicken Theorem** The chicken with the largest outdegree in an n-chicken tournament is a king.

Proof. We use proof by contradiction. Let v be a node in a tournament graph $G = (V, E)$ with maximum outdegree and suppose that v is not a king. Let $S := \{u \mid (v, u) \in E\}$ be the set of chickens that chicken v pecks. Then $\text{outdeg}\, v = |S|$.

Since v is not a king, there is a king chicken $x \notin S$ and that is not pecked by any chicken in S. Since for any pair of chickens, one pecks the other, this means the x pecks v as well as every chicken in S. This means that

$$\text{outdeg}\, x = |S| + 1 > \text{outdeg}\, v\,, \tag{7.15}$$

contradicting to the fact that v has the largest out degrees. □

The above theorem means that if the player with the most victories is defeated by another player x, then at least he/she defeats some third player that defeats x. However, for some tournaments, it is possible that some player with fewer victories is also the king.

7.4 Communication Networks

7.4.1 Routing Problems

In communication networks, vertices represent computers, processors, and switches; edges will represent wires, fiber, or other transmission lines through which data flows. we consider aim to transmit packets of data between vertices.

DEFINITION 7.34 **Packet** Some roughly fixed-size quantity of data, say 256 Bytes or 4096 Bytes.

DEFINITION 7.35 **Terminal** Sources and destinations for packets of data. Terminals are represented as squares in the diagram.

DEFINITION 7.36 **Switch** A switch is used to direct packets through the network. A switch receives packets on incoming edges and relays them forward along the outgoing edges. Switches are represented as squares in the diagram.

DEFINITION 7.37 **Permutation** A permutation is a function $\pi\ \{0, 1, \ldots, n-1\} \mapsto \{0, 1, \ldots, n-1\}$ such that no two numbers are mapped to the same value, i.e.,

$$\forall i, j.\ \pi(i) = \pi(j) \Rightarrow i = j\,. \tag{7.16}$$

DEFINITION 7.38 **Routing Problem** Suppose n is a power of 2. Communication networks are supposed to get packets from inputs to outputs, with each packet entering the network at its own input switch $i \in \{0, 1, \ldots, n-1\}$ and arriving at its own output switch $\pi(i)$.

DEFINITION 7.39 **Routing** A routing P that solves a routing problem π, is a set of paths from each input to its specified output. That is, P is a set of n paths, where p_i goes from input i to output $\pi(i)$.

7.4.2 Four Parameters for Communication Networks

DEFINITION 7.40 **Diameter** Maximum length of any shortest path between an input and an output.

DEFINITION 7.41 **Latency** The length of the longest path in a routing.

DEFINITION 7.42 **Network Latency** The largest routing latency among these optimal routing.

Network latency will equal network diameter if routings are always chosen to optimize delay, but it may be significantly larger if routings are chosen to optimize something else. For the networks we consider below, paths from input to output are uniquely determined or all paths are the same length, so network latency will always equal network diameter.

DEFINITION 7.43 **Switch Size** A $k \times k$ switch means that the switch has k incoming edges and k outgoing edges.

DEFINITION 7.44 **Switch Count** Number of switches.

DEFINITION 7.45 **Congestion** The largest number of paths in P that pass through a single switch.

DEFINITION 7.46 **Network Congestion** For each routing problem π for the network, we assume a routing is chosen that optimizes congestion, that is, that has the minimum congestion among all routings that solve π. Then the largest congestion that will ever be suffered by a switch will be the network congestion among these optimal routings. I.e.,

$$\max_{\pi} \min_{P} \text{congestion}(P). \tag{7.17}$$

7.4.3 Complete Binary Tree

See Fig. 7.3. There is a unique path between every pair of vertices in a tree.

- Diameter: $2(\lg n + 1) = 2\lg n + 2$.
- Switch size: 2×2 and 3×3.
- Switch count: $1 + 2 + 4 + \cdots + n = \sum_{i=1}^{n} 2^i = 2n - 1$.
- Congestion: n. The worst permutation is

$$\pi(i) = n - 1 - i. \tag{7.18}$$

Then every packet would have to follow a path through the root switch.

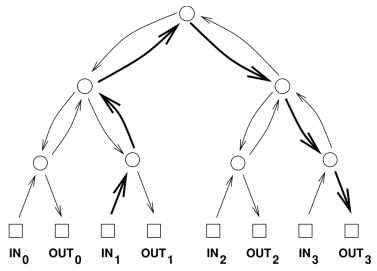

Figure 7.3
A complete binary tree for 4 inputs and 4 outputs.

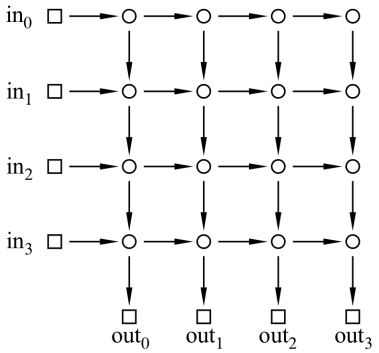

Figure 7.4
A 2-D array for 4 inputs and 4 outputs.

7.4.4 2-D Array

See Fig. 7.4.

- Diameter: $2n$.
- Switch size: 2×2, 1×2 and 2×1.
- Switch count: n^2.
- Congestion: 2.

Proof. First, we show that the congestion is ≤ 2. Let p_i goes to the right from input i to column $\pi(i)$ and then goes down to output $\pi(i)$. Thus, the switch in row i and column j transmits ≤ 2 packets: the packet originating at input i and the packet destined for output j.

Next, we show that the congestion is ≥ 2. This follows because in any routing problem π where $\pi(0) = 0$ and $\pi(n-1) = n-1$, two packets must pass through the lower left switch. □

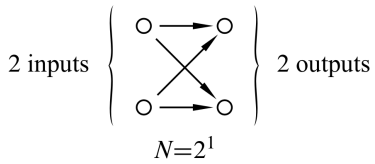

Figure 7.5

F_1, the butterfly switches with $n = 2^1$.

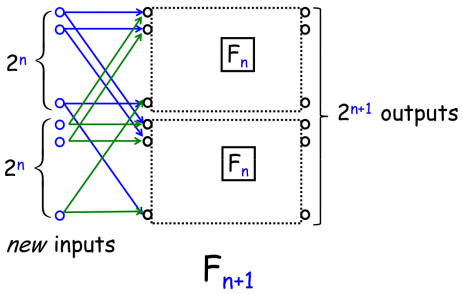

Figure 7.6

F_{k+1}, the butterfly switches with $n = 2^{k+1}$.

7.4.5 Butterfly

We describe the butterfly network by a recursive defined data type F_k with $n = 2^k$ input and output switches, omitting the terminals.

- Base case: F_1 with 2 input and output switches, see Fig. 7.5.
- Constructor step: We construct F_{k+1} with 2^{k+1} input and output switches out of two F_n nets, see Fig. 7.6. For $i = 0, 1, 2, \ldots, 2^k - 1$, the i-th and $(2^k + i)$-th input switches are each connected to the i-th input switches of each of two F_n components.

See Fig. 7.7 for an example.

Let us label the rows in binary so that the label on row i is the binary number $b_1 b_2 \ldots b_{\lg n}$ that represents the integer i. Between the inputs and outputs, there are $\lg n + 1$ levels of switches, numbered from 0 to $\lg n$. Each level consists of a column of n switches, one per row. Thus, each switch in the network is uniquely identified by a sequence

$$(b_1, b_2, \ldots, b_{\lg n}, l)\,, \tag{7.19}$$

where $b_1 b_2 \ldots b_{\lg n}$ is the switch's row in binary and l is the switch's level.

The connection pattern is expressed below. There are directed edges from each switch to two switches in the next level. One edges leads to the switch in the same row, and the

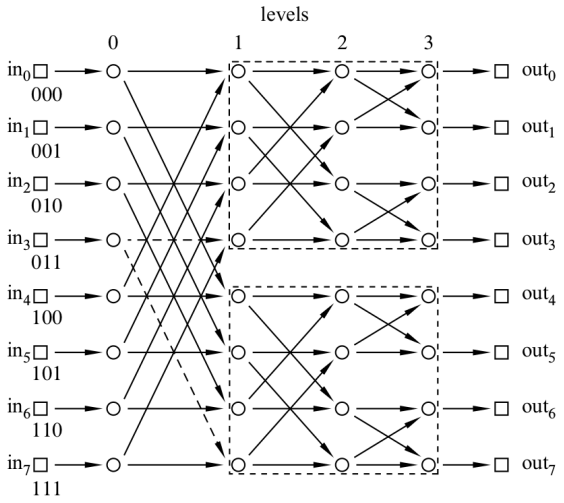

Figure 7.7
A butterfly for 8 inputs and 8 outputs.

other edge leads to the switch in the row obtained by inverting the $(l+1)$-th bit b_{l+1}.

$$(b_1, b_2, \ldots, b_{l+1}, \ldots, b_{\lg n}, l) \to (b_1, b_2, \ldots, b_{l+1}, \ldots, b_{\lg n}, l+1)\,, \tag{7.20}$$

$$(b_1, b_2, \ldots, b_{l+1}, \ldots, b_{\lg n}, l) \to (b_1, b_2, \ldots, \overline{b_{l+1}}, \ldots, b_{\lg n}, l+1)\,. \tag{7.21}$$

Suppose we want to send a package from input $x_1 x_2 \ldots x_{\lg n}$ to $y_1 y_2 \ldots y_{\lg n}$. (Here we are specifying the input and output numbers in binary.) The plan is to "correct" the first bit on the first level, correct the second bit on the second level, and so forth.

$$(x_1, x_2, \ldots, x_{\lg n}, 0) \to (y_1, x_2, \ldots, x_{\lg n}, 1) \tag{7.22}$$

$$\to (y_1, y_2, \ldots, x_{\lg n}, 2) \tag{7.23}$$

$$\to \ldots \tag{7.24}$$

$$\to (y_1, y_2, \ldots, y_{\lg n}, \lg n)\,. \tag{7.25}$$

In fact, this is the only path from the input to the output.

- Diameter: $\lg n + 2$.
- Switch size: 2×2, 2×1, and 1×2.
- Switch count: $n(\lg n + 1) = n \lg n + n$.
- Congestion: $\sqrt{n}$ if n is an even power of 2 or $\sqrt{\frac{n}{2}}$ if n is an odd power of 2.

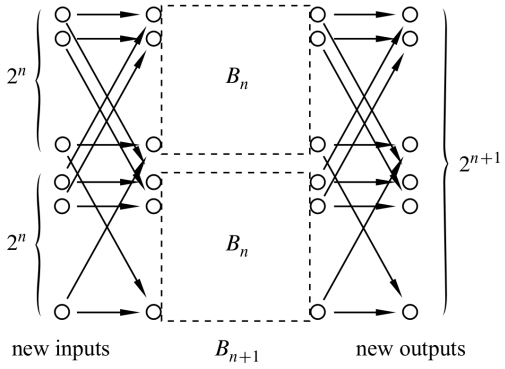

Figure 7.8

B_{k+1}, the Beneš network with $n = 2^{k+1}$.

7.4.6 Beneš Network

Beneš Network is composed by two butterfly networks back-to-back. We recursively define data type B_k with $n = 2^k$ input and output switches as follows

- Base case: B_1 with 2 input and output switches is exactly the same as F_1 in Fig. 7.5.
- Constructor step: We construct B_{k+1} with 2^{k+1} input and output switches out of two B_n nets, see Fig. 7.8. For $i = 0, 1, 2, \ldots, 2^k - 1$, the i-th and $(2^k + i)$-th input switches are each connected to the i-th input switches of each of two B_n components. In addition, the i-th and $(2^k + i)$-th new output switches are connected to the same two switches, namely, to the i-th output switches of each of two B_n components.
  See Fig. 7.9 for an example.
- Diameter: $2 \lg n - 1$.
- Switch size: 2×2, 2×1, and 1×2.
- Switch count: $2n \lg n$.
- Congestion: 1.

DEFINITION 7.47 **Constraint Graph** If two packets must pass through different subnetworks, then there is an edge between them.

LEMMA 7.48 If the edges of a graph can be grouped into two sets such that every vertex has at most 1 edge from each set incident to it, then the graph is 2-colorable.

THEOREM 7.49 The congestion of the n-input Beneš network is 1.

Proof. By induction on k where $n = 2^k$. Let the inductive hypothesis be $P(k) :=$ the congestion of B_k is 1.
Base case: $B_1 = F_1$. The unique routings in F_1 have congestion 1.

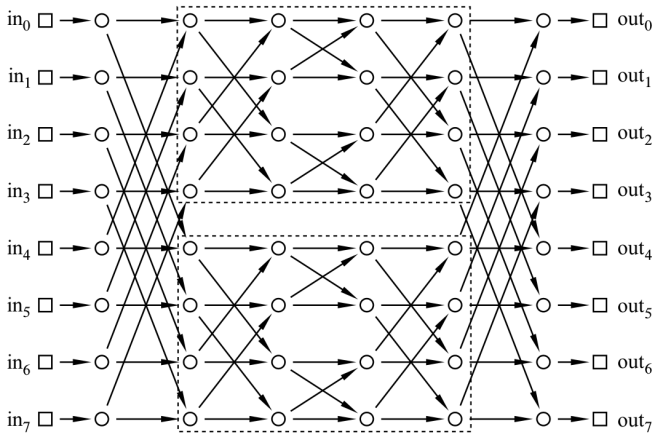

Figure 7.9
A Beneš network for 8 inputs and 8 outputs.

Base case: Suppose $P(k-1)$ is true. Let π be an arbitrary permutation of $\{0,1,\ldots,n-1\}$. We cannot route both packet i and packet $\frac{n}{2}+i$ through the same subnetwork, since that would cause two packets to collide at a single switch at the input of the subnetwork, resulting a congestion. Also, if $\pi(j_1)=i$ and $\pi(j_2)=\frac{n}{2}+i$ for some i, then we cannot route both packet j_1 and packet j_2 through the same subnetwork, since that would cause two packets to collide at a single switch at the output of the subnetwork, resulting a congestion. We represent these two kinds of constrains in a graph.

Let G be the graph whose vertices are packets numbers $0,1,\ldots,n-1$ and whose edges come from the union of these two sets:

$$E_1 := \left\{(u,v) \mid |u-v| = \frac{n}{2}\right\}, \tag{7.26}$$

$$E_2 := \left\{(u,v) \mid |\pi(u)-\pi(v)| = \frac{n}{2}\right\}. \tag{7.27}$$

Now any vertex v, is incident to at most 2 edges: a unique edge in E_1 and a unique edge in E_2. So according to the lemma, there is a 2-coloring for the vertices of G. Now route packages of one color through the upper subnetwork and packets in the other to the lower subnetwork.

Since for each edges in E_1, one vertex goes to the upper subnetwork and the other to the lower subnetwork, there will not be any conflicts in the first level. Since for each edges in E_2, one vertex comes from the upper subnetwork and the other from the lower subnetwork, there will not be any conflicts in the last level. We can complete the routing with each subnetwork by the induction hypothesis $P(k-1)$. □

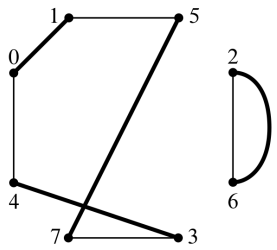

Figure 7.10
Constraint graph of B_3.

Table 7.1
Four communication networks.

Network	Diameter	Switch Size	Switch Count	Congestion
Complete binary tree	$2\lg n+2$	3×3	$2n-1$	n
2-D array	$2n$	2×2	n^2	2
Butterfly	$\lg n+2$	2×2	$n\lg n+n$	$\sqrt{n}$ or $\sqrt{\frac{n}{2}}$
Benes	$2\lg n+1$	2×2	$2n\lg n$	1

For example, consider the following permutation routing problem:

$$\pi(0)=1,\pi(1)=5,\pi(2)=4,\pi(3)=7,\pi(4)=3,\pi(5)=6,\pi(6)=0,\pi(7)=2 \quad (7.28)$$

Then the constraint graph of can be seen in Fig. 7.10. We intend this to be a single graph, so the two lines between 2 and 6 still signify a single edge. The coloring corresponds to a solution to the routing problem.

7.4.7 Summary

The Tab. 7.1 summaries the four communication networks.

8 Binary Relations, Partial Orders, and DAG

8.1 Digraph and Binary Relations

8.1.1 Relational Properties

DEFINITION 8.1 **Reflexivity** Every vertex in R has a self-loop.

$$\forall x \in A.\ xRx\,. \tag{8.1}$$

DEFINITION 8.2 **Irreflexivity** There are no self-loops in R.

$$\neg(\exists x \in A.\ xRx)\,. \tag{8.2}$$

DEFINITION 8.3 **Symmetry** If $x \to y$ in R, then there is an edge $y \to x$.

$$\forall x, y \in A.\ xRy \Rightarrow yRx\,. \tag{8.3}$$

DEFINITION 8.4 **Asymmetry** There is at most one directed edge between any two vertices in R, and there are no self-loops.

$$\forall x, y \in A.\ xRy \oplus yRx\,. \tag{8.4}$$

DEFINITION 8.5 **Antisymmetry** There is at most one directed edge between any two distinct vertices in R, but there may be self-loops.

$$\forall x \neq y \in A.\ xRy \oplus yRx\,. \tag{8.5}$$

Equivalently,

$$\forall x, y \in A.\ (xRy \wedge yRx) \Rightarrow x = y\,. \tag{8.6}$$

DEFINITION 8.6 **Transitivity** If there is a positive length path from u to v, then there is an edge from u to v.

$$\forall x, y, z \in A.\ (xRy \wedge yRz) \Rightarrow xRz\,. \tag{8.7}$$

DEFINITION 8.7 **Linear** Given any two vertices in R, there is an edge in one direction or the other between them.

$$\forall x, y \in A.\ xRy \vee yRx\,. \tag{8.8}$$

8.1.2 Examples

LEMMA 8.8 For any digraph G, the walk relation G^* is reflexive.

LEMMA 8.9 For any digraph G, the walk relation G^* and G^+ are transitive.

Some examples are summarized as in Tab. 8.1.

Table 8.1
Some examples of properties of relations.

R	Reflexivity	Symmetry	Antisymmetry	Transitivity	Comment
$x \equiv y \pmod{n}$	True	True	False	True	Equivalence
$x \mid y$	True	False	True	True	Partial Order
$x \leq y$	True	False	True	True	Partial Order

8.1.3 Equivalence Relations

DEFINITION 8.10 **Equivalence Relation** R is an equivalence relation if R is reflexive, symmetric, and transitive.

Examples of equivalence relations: $=, \equiv \pmod{n}$, same size, same color.

DEFINITION 8.11 $\equiv_f$ If $f\colon A \mapsto B$ is a total function, define a relation $\equiv_f$ by the rule:

$$a \equiv_f a' ::= f(a) = f(a'). \tag{8.9}$$

$\equiv_f$ is reflexive, symmetric and transitive because these are properties of equality. That is, $\equiv_f$ is an equivalence relation.

DEFINITION 8.12 **Equivalence Class** Given an equivalence relation R: $A \mapsto A$, the equivalence class, $[a]_R$ of an element $a \in A$ is the set of all elements of A equivalent to a by R. Namely,

$$[a]_R ::= \{a' \in A \mid aRa'\}. \tag{8.10}$$

In other words, $[a]_R$ is the image $R(a)$.

THEOREM 8.13 R is an equivalence relation if R equals the in-the-same-block relation for some partition of $\operatorname{dom} R$.

THEOREM 8.14 **An Equivalence Relation is the Same as a Partition** The equivalence classes of any equivalence relation R on a set A form a partition of A, and any partition of A determines an equivalence relation on A for which the sets in the partition are the equivalence classes.

8.2 Partial Orders

8.2.1 Weak Partial Orders

DEFINITION 8.15 **Weak Partial Order** R is a weak partial order $\preceq$ if R is reflexive, antisymmetric, and transitive. Examples of strict partial orders are: $\leq, \subseteq, |$.

THEOREM 8.16 Every weak partial order $\preceq$ is isomorphic to the subset relation $\subseteq$ on a collection of sets. We simply represent each element $v \in V$ by the set

$$v \leftrightarrow \{v' \in V \mid v' \preceq v\}, \forall v\,. \tag{8.11}$$

8.2.2 Strict Partial Orders

DEFINITION 8.17 **Strict Partial Order** R is a strict partial order $\prec$ if R is irreflexive, antisymmetric, and transitive. Examples of strict partial orders are: $<, \subset$.

THEOREM 8.18 Every strict partial order $\prec$ is isomorphic to the proper subset relation $\subset$ on a collection of sets. We simply represent each element $v \in V$ by the set

$$v \leftrightarrow \{v' \in V \mid v' \prec v\}, \forall v\,. \tag{8.12}$$

8.2.3 Total Orders

DEFINITION 8.19 **Comparable** We say that two elements u and v are comparable iff $a \preceq b \vee b \preceq a$, i.e, one vertex can be reached from the other.

DEFINITION 8.20 **Total Order/Linear Order** A total order $\preceq_T$ is a partial order in which every pair of distinct elements is comparable.

DEFINITION 8.21 **Total Preorder** A total relation that is transitive, but not necessarily either symmetric or antisymmetric.

Examples of total orders: $<, \leq$. On the other hand, the subset relation is not total, since, for example, any two different finite sets of the same size will be incomparable under $\subseteq$.

8.2.4 Product Orders

Products of relations preserve the properties of transitivity, reflexivity, irreflexivity, and antisymmetry. If R and S both have one of these properties, then so does $R \times S$. This implies that if R and S are both partial orders, then so is $R \times S$. On the other hand, the property of being a total order is not preserved.

8.3 Posets and DAGs

8.3.1 Posets and DAGs

DEFINITION 8.22 **Partially Ordered Set (Poset)** Given a weak partial order $\preceq$ on a set A, the pair $(A, \preceq)$ is called a partially ordered set or poset. In terms of graph theory, a poset is simply the directed graph $G = (A, \preceq)$ with vertex set A and edge set $\preceq$.

THEOREM 8.23 A poset has no directed cycles other than self-loops.

Proof. We use proof by contradiction. Let $(A, \preceq)$ be a poset. Suppose that there exists $n \geq 2$ distinct elements $a_1, a_2, \ldots, a_n$ such that

$$a_1 \preceq a_2 \preceq \cdots \preceq a_n \preceq a_1 \,. \tag{8.13}$$

Since $a_1 \preceq a_2 \wedge a_2 \preceq a_3$, transitivity implies $a_1 \preceq a_2$, and a routine induction argument establishes that $a_1 \preceq a_n$. Since we know that $a_n \preceq a_1$, antisymmetry implies that $a_1 = a_n$, contradicting the assumption that $a_1, a_2, \ldots, a_n$ are distinct. □

DEFINITION 8.24 **Directed Acyclic Graph (DAG)** A directed acyclic graph (DAG) is a directed graph with no cycles.

COROLLARY 8.25 Deleting the self-loops from a poset leaves a directed graph without cycles, which makes it a DAG.

Does every DAG correspond to a poset? Any DAG must satisfies the antisymmetry property but it may not satisfy the transitivity property. So we need to modify the DAG according to the transitive closure.

DEFINITION 8.26 **Transitive Closure** Given a digraph $D = (V, E)$, the transitive closure of D is the digraph $\hat{D} = (V, \hat{E})$ where

$$\hat{E} := \{u \to v \mid uD^+v\} \,. \tag{8.14}$$

That is to say, $(u, v) \in E$ if there is a directed path of positive length from u to v in G.

DEFINITION 8.27 **Hasse Diagram** A Hasse diagram for a poset $(A, \preceq)$ is a digraph with vertex set A and edge set $\preceq$ minus all self-loops and edges implied by transitivity.

8.3.2 Topological Sort

DEFINITION 8.28 **Topological Sort** A topological sort of a poset $(A, \preceq)$ is a total order $(A, \preceq_T)$ which is consistent with partial order:

$$x \preceq y \Rightarrow x \preceq_T y \,. \tag{8.15}$$

In other words, A topological sort of a finite DAG is a list of all the vertices such that each vertex v appears earlier in the list than every other vertex reachable from v. There might be several total orders that are consistent with the partial order.

DEFINITION 8.29 **Minimum** In a DAG D, a vertex v is minimum iff every other vertex in D is reachable from v.

DEFINITION 8.30 **Maximum** In a DAG D, a vertex v is maximum iff it is reachable from all other vertices in D.

A finite chain is said to end at its maximum element.

DEFINITION 8.31 **Minimal** In a DAG D, a vertex v is minimal iff v is not reachable from any other vertex in D, i.e,

$$\exists u \neq v.\ u \preceq v\,. \tag{8.16}$$

DEFINITION 8.32 **Maximal** In a DAG D, a vertex v is maximal iff v can not reach any other vertex in D, i.e,

$$\exists u \neq v.\ v \preceq u\,. \tag{8.17}$$

A DAG may have no minimum element but lots of minimal elements.

LEMMA 8.33 Every finite poset has a minimal element.

Proof. Let $(A, \preceq)$ be an arbitrary poset. Let $a_1, a_2, \ldots, a_n$ be a maximum length sequence of distinct elements in A such that

$$a_1 \preceq a_2 \preceq \cdots \preceq a_n\,. \tag{8.18}$$

The existence of such a maximum-length sequence follows from the Well Ordering Principle and the fact that A is finite.

Now $a_0 \preceq a_1$ cannot hold for any element $a_0 \in A$ not in the chain, since the chain already has maximum length, and $a_i \preceq a_1$ cannot hold for any $i \geq 2$, since that would imply a cycle and no cycles exist in a poset. Therefore a_1 is a minimal element. □

THEOREM 8.34 Every finite DAG has a topological sort.

Proof. We use proof by induction. Let the inductive hypothesis be $P(n) :=$ every n-element poset has a topological sort.

Base case: Every 1-element poset is already a total order and thus is its own topological sort, so $P(1)$ is true.

Inductive step: Assume $P(n)$ is true. Let $(A, \preceq)$ be an $(n+1)$-element poset. There exists a minimal element $a_0 \in A$. Remove a and all pairs in $\preceq$ involving a_0 to obtain an n-element poset $(A', \preceq')$. This has a total order $(A', \preceq'_T)$ by the assumption $P(n)$.

Now we construct a total order $(A, \preceq_T)$ by adding a_0 back as an element smaller than all the others, namely,

$$\preceq_T := \preceq'_T \cup \{(a_0, a) \mid a \in A\} \tag{8.19}$$

All that remains is to check that this total order is consistent with the original partial order. That is, we need to show that

$$x \preceq y \Rightarrow x \preceq_T y\,. \tag{8.20}$$

There are two cases:

Case 1: $x = a_0$, then $a_0 \preceq_T y$, since $\forall a \in A.\ a_0 \preceq_T a$.
Case 2: $x \neq a_0$, then $y \neq a_0$, since a_0 is the minimal element in the partial order $\preceq$. Thus, $x, y \in A'$ and so $x \preceq' y$. This means that $x \preceq'_T y$, since $\preceq'_T$ is a topological sort of the partial order $\preceq'$. □

To perform topological sort, we iteratively pick one minimal element, and remove the corresponding node and edges from the graph. Continuing in this way until all elements have been picked, then the sequence of elements in the order they were picked will be a topological sort.

8.4 Parallel Task Scheduling

8.4.1 Scheduling Problem

DEFINITION 8.35 **Scheduling Problem** In a scheduling problem, there is a set of tasks, along with a set of constraints specifying that starting certain tasks depends on other tasks being completed beforehand. We can map these sets to a digraph, with the tasks as the nodes and the direct prerequisite constraints as the edges. If $u \to v \in E$, then task u must be taken before task v.

The most basic task in scheduling is finding an order in which to perform all the tasks, one at a time, while respecting the dependency constraints.

8.4.2 Parallel Schedule

For scheduling problems, topological sorting provides a way to execute tasks one after another while respecting those dependencies. But what if we have the ability to execute more than one task at the same time?

For simplicity, let's say all the tasks take the same amount of time and all the processors are identical and the number of available processors is unlimited. Our goal should be to minimize the total time to complete all the tasks.

DEFINITION 8.36 **Parallel Schedule** A parallel schedule for a DAG D is a partition of V into blocks $A_0, A_1, \ldots$ such that when $i < j$, no vertex in A_i is reachable from any vertex in A_j. The block A_t is called the set of elements scheduled at step t, and the time of the schedule is the number of blocks. The maximum number of elements scheduled at any step is called the number of processors required by the schedule.

8.4.3 Chain and Antichain

DEFINITION 8.37 **Chain** A chain in a DAG is a sequence $v_1 \preceq v_2 \preceq \cdots \preceq v_T$ such that any two of them are comparable. The length of the chain is T, the number of elements in the chain.

DEFINITION 8.38 **Antichain** An antichain in a DAG is a set of vertices such that no two elements in the set are comparable, i.e., no walk exists between any two different vertices in the set.

DEFINITION 8.39 **Critical Path** A longest chain.

DEFINITION 8.40 **Critical Path to** v A chain ending at v with largest number of vertices in that chain.

DEFINITION 8.41 **Depth** The number of nodes preceding v in the critical path to v is called the depth of v. We denote it as $\operatorname{depth} v$. In particular, the minimal elements are precisely the elements with depth 0.

8.4.4 Minimum Time Scheduling

There is a very simple schedule that completes every task in its minimum number of steps: in each unit of time, we should do all minimal items. That is, we schedule all the elements of depth t at step t.

THEOREM 8.42 A minimum time schedule for a finite DAG D consists of the sets $A_0, A_1, \ldots$ where

$$A_t := \{v \in V \mid \operatorname{depth} v = t\}. \tag{8.21}$$

COROLLARY 8.43 For any DAG D, there is a legal parallel schedule that runs all the tasks with T step, where T is length of critical path, and V can be partitioned into T antichains.

LEMMA 8.44 [Dilworth] For all $t \in \mathbb{Z}^+$, every DAG with n vertices must have
- Either a chain of length $> t$.
- Or an antichain of size $\geq \frac{n}{t}$.

Proof. By contradiction. Assume that the longest chain has length $\leq t$ and the longest antichain has length $< \frac{n}{t}$. Then the n element can be partitioned into *let* antichains. Hence, there are fewer than $t \cdot \frac{n}{t} = n$ elements in the poset, which is a contradiction. □

COROLLARY 8.45 Every DAG with n vertices must have
- Either a chain of length $\sqrt{n}$.
- Or an antichain of size $\sqrt{n}$.

Proof. Let $t = \sqrt{n}$. □

9 Simple Graphs

9.1 Simple Graph Basis

DEFINITION 9.1 **Simple Graph** A simple graph $G = (V, E)$ consists of a set $V \neq \varnothing$ called the **vertex set** of G, and a set E called the **edge set** of G. An element of V is called a **vertex/node**. An element of E is an **undirected edge/edge**. An undirected edge has two vertices $u \neq v$ called its **endpoints**. Such an edge can be represented by $\{u, v\}$ or $u - v$. Simple graphs model relationships that are symmetric.

Note there is no self-loop nor multiple edges between two vertices in a simple graph.

DEFINITION 9.2 **Weighted Graph** A weighted graph consists of a graph $G = (V, E)$ and a weight function w: $E \mapsto \mathbb{R}$.

DEFINITION 9.3 **Adjacency Matrix** Given an n-node graph $G = (V, E)$ where $V = \{v_i\}_{i=1}^n$, the adjacency matrix for G is the $n \times n$ matrix $\boldsymbol{A}$ where

$$A_{ij} = \mathbb{I}\{\{v_i, v_j\} \in E\}\,. \tag{9.1}$$

If G is a weighted graph with edge weights given by w: $E \mapsto \mathbb{R}$, then the adjacency matrix for G is $\boldsymbol{A}$ where

$$A_{ij} = \mathbb{I}\{\{v_i, v_j\} \in E\} w(\{v_i, v_j\})\,. \tag{9.2}$$

9.1.1 Vertex Adjacency and Degrees

DEFINITION 9.4 **Adjacent** Two vertices in a simple graph are said to be adjacent iff they are the endpoints of the same edge.

DEFINITION 9.5 **Incident** An edge is said to be incident to each of its endpoints.

DEFINITION 9.6 **Degree** The number of edges incident to a vertex v is called the degree of the vertex and is denoted by $\deg v$. Equivalently, the degree of a vertex is the number of vertices adjacent to it.

LEMMA 9.7 [Handshaking Lemma] The sum of the degrees of the vertices in a graph equals twice the number of edges.

Proof. Every edge contributes two to sum of the degrees, one for each of its endpoints. □

There an an even number of vertices with odd degree.

9.1.2 Walks, Paths, and Cycles

DEFINITION 9.8 **Walk** A walk in a simple graph G is an alternating sequence of vertices $v_0, v_1, v_2, \ldots, v_k$ such that $\forall i \in \mathbb{Z}_k$. $v_i - v_{i+1} \in E$. The walk is said to start at v_0, and end at v_k, and the length of a walk is the total number of occurrences of edges in it, which is defined to be k.

DEFINITION 9.9 **Path** The walk is a path iff all the v_i's are different, that is, if $i \neq j \Rightarrow v_i - v_j$.

DEFINITION 9.10 **Closed Walk** A walk that begins and ends at the same vertex.

A single vertex counts as a length zero closed walk as well as a length zero path.

DEFINITION 9.11 **Cycle** A closed walk of length three or more whose vertices are distinct except for the beginning and end vertices.

Note that in contrast to digraphs, we do not count length two closed walks as cycles in simple graphs. Also, there are no closed walks of length one, since simple graphs do not have self loops.

DEFINITION 9.12 **Subgraph** A graph $G' = (V', E')$ is said to be a subgraph of a graph $G = (V, E)$ if $V' \subseteq V \wedge E' \subseteq E$.

- An empty graph on n nodes will be a subgraph of an L_n with the same set of nodes.
- L_n is a subgraph of C_n.
- C_n is a subgraph of K_n.

DEFINITION 9.13 **Induce** The subgraph of $G = (V, E)$ induced by V' is the graph $G' = (V', E')$ where

$$E' := \{(u, v) \in E \mid u, v \in V'\}. \tag{9.3}$$

DEFINITION 9.14 **Contraction** The contraction of an undirected graph $G = (V, E)$ by an edge $e = \{u, v\}$ is a graph $G' = (V', E')$ where

$$V' := (V - \{u, v\}) \cap \{x\}, \tag{9.4}$$

where x is a new vertex. The set of edges E' is formed from E by deleting the edge e and, for each vertex x' adjacent to u or v, deleting whichever of $\{u, x'\}$ or $\{v, x'\}$ is in E and adding the new edge $\{x, x'\}$. In effect, u and v are "contracted" into a single vertex.

9.1.3 Some Common Graphs

DEFINITION 9.15 **Complete Graph K_n** A complete graph K_n has n vertices and an edge between every two vertices, for a total of $n(n-1)/2$ edges.

DEFINITION 9.16 **Empty Graph** The empty graph has no edges at all.

DEFINITION 9.17 **Line Graph L_n** An n-node graph containing $n-1$ edges in sequence is known as a line graph L_n:

$$V = \{v_1, v_2, \ldots, v_n\}, \tag{9.5}$$

$$E = \{v_i - v_{i+1} \mid 1 \leq i \leq n-1\}. \tag{9.6}$$

DEFINITION 9.18 **Cycle Graph C_n** If we add the edge $v_n - v_1$ to the line graph L_n, we get a graph called a length-n cycle C_n.

9.2 Isomorphism

DEFINITION 9.19 **Isomorphism** An isomorphism between graphs $G = (V, E)$ and $G' = (V', E')$ is a bijection $f \;\; V \mapsto V'$ such that

$$\forall u, v \in V.\; u - v \in E \Leftrightarrow f(u) - f(v) \in E'. \tag{9.7}$$

DEFINITION 9.20 **Isomorphic** Two graphs are isomorphic when there is an isomorphism between them.

COROLLARY 9.21 Isomorphism is an equivalence relation.

- f is an isomorphism from G to $G' \Rightarrow f^{-1}$ is an isomorphism from G' to G.
- Isomorphism is transitive because the composition of isomorphisms is an Isomorphism.

Isomorphism preserves the connection properties of a graph, abstracting out:

- What the vertices are called.
- What they are made out of.
- Where they appear in a drawing of the graph.

DEFINITION 9.22 **Preserved Under Isomorphism** A property of a graph is said to be preserved under isomorphism if whenever G has that property, every graph isomorphic to G also has that property. The examples of those properties are

- Number of nodes.
- Number of edges.
- Degree distributions.

Proving nonisomorphism: If some property preserved by isomorphism differs for two graphs, then they are not Isomorphic.

No one has yet found a procedure for determining whether two graphs are isomorphic that is guaranteed to run in polynomial time on all pairs of graphs. However, it is generally easy in practice to decide whether two graphs are isomorphic.

9.3 Coloring

There are lots of situations in which edges will correspond to conflicts between nodes. One reason coloring problems frequently arise in practice is because scheduling conflicts are so common.

9.3.1 Graph Coloring Problem

DEFINITION 9.23 **Graph Coloring Problem** Given a graph G assign a color to each vertex so that adjacent vertices get different colors.

DEFINITION 9.24 **k-Colorable** A graph G is k-colorable if it has a valid coloring that uses $\leq k$ colors.

DEFINITION 9.25 **Chromatic Number $\chi(G)$** The minimum value of k for which a graph G is k-colorable.

In general, there is no fast algorithms to color a graph with a fixed number of colors. In fact, it is easy to check if a coloring works, but it seems really hard to find it.

COROLLARY 9.26 An even-length closed cycle is 2-colorable.

$$\forall i \in \mathbb{N}^+.\ \chi(C_{2i}) = 2\,. \tag{9.8}$$

An odd-length cycle requires 3 colors.

$$\forall i \in \mathbb{N}.\ \chi(C_{2i+1}) = 3\,. \tag{9.9}$$

No two vertices can have the same color in a complete graph.

$$\forall n \in \mathbb{N}.\ \chi(K_n) = n\,. \tag{9.10}$$

9.3.2 2-Colorability

THEOREM 9.27 The following graph properties are equivalent:
- The graph is bipartite.
- The graph is 2-colorable.
- The graph does not contain an odd length cycle.
- The graph does not contain an odd length closed walk.

Proof. **1 ⇒ 2**: Assume that $G = (V, E)$ is a bipartite graph. We can use one color for all the nodes in V_L and a second color for all the nodes in V_R. Hence $\chi(G) = 2$.
2 ⇒ 3: Let $G = (V, E)$ be a 2-colorable graph and $v_0, v_1, \ldots, v_k$ be any cycle in G. Since $\{v_i, v_{i+1}\} \in E, i \in [0, k)$, v_i and v_{i+1} must be differently colored. Hence $v_0, v_2, v_4, \ldots$ have

one color and $v_1, v_3, v_5, \ldots$ have the other color. Since it is a cycle, $v_k = v_0$, and they must have the same color, so k must be even. This means that the cycle has even length.

3 ⇒ 4: We use proof by contradiction. Assume that G is graph that does not contain any cycles with odd length, and G contains a closed walk with odd length. Let $p := v_0, v_1, \ldots, v_k$ be the shortest closed walk with odd length. Since G has no odd-length cycles, p cannot be a cycle.

Hence there exists $0 \leq i < j < k$ such that $v_i = v_j$. This means that p is the union of two closed walk

$$p = v_0, v_1, \ldots, v_i, v_{j+1}, v_{j+2}, \ldots, v_k \cup v_i, v_{i+1}, \ldots, v_j \,. \tag{9.11}$$

Since p has odd length, one of these closed walk must also have odd length. This contradicting to the assumption that p is the minimum length closed walk with odd-length.

4 ⇒ 1: We use proof by contradiction. Assume that G is a graph without any odd length closed walk, and G is not bipartite. There must be a connected component $G' = (V', E')$ that is not bipartite.

Let $v \in V'$. For every node $u \in V'$, let $\operatorname{dist} u :=$ the length of the shortest path from u to v in G', and $\operatorname{dist} v = 0$. Partition V' into sets V'_L and V'_R such that

$$V'_L = \{u \mid \operatorname{rem}(\operatorname{dist} u, 2) = 0\}\,, \tag{9.12}$$

$$V'_R = \{u \mid \operatorname{rem}(\operatorname{dist} u, 2) = 1\}\,. \tag{9.13}$$

Since G' is not bipartite, there must be a pair of adjacent nodes u_1 and u_2 that are both in V'_L or both in V'_R. Let e denote the edge incident to u_1 and u_2. Let p_1 denote a shortest path in G' from u_1 to v, similarly for p_2. Then p_1 and p_2 must both have even length or odd length. In either case, the union of p_1, p_2, and e form a closed walk, which is a contradiction. □

9.3.3 Coloring Bound

THEOREM 9.28 A graph with maximum degree at most d is $(d+1)$-colorable.

Proof. We use proof by induction on the number of vertices n. Let the inductive hypothesis be $P(n) :=$ any n-vertex graph with maximum degree $\leq d$ is $(d+1)$-colorable.

Base case: $n = 1$, then a 1-vertex graph has maximum degree 0 and is 1-colorable, so $P(1)$ is true.

Inductive step: Assume that $P(n)$ is true, and let $G = (V, E)$ where $|V| = n + 1$ and the maximum degree is at most d. Remove a vertex v and all edges incident to v leaving an v-vertex subgraph G'. The maximum degree of G' is at most d, and so G' is $(d+1)$-colorable by our assumption $P(n)$.

Now add back vertex v. We can assign v a color (from the set of $d+1$ colors) that is different from all its adjacent vertices, since there are at most d vertices adjacent to v and so at least one of the $d+1$ colors is still available. Therefore, G is $d+1$-colorable.

This completes the inductive step, and the theorem follows by induction. □

Algorithm 10 Greedy Coloring Algorithm

Input: $G = (V, E)$; A set of colors C.
Output: A valid coloring of G.

Order the nodes $v_1, v_2, \ldots, v_n$.
Order the colors $c_1, c_2, \ldots$.
for $i = 1$ **to** n
  Assign v_i with the lowest legal color.

See Alg. 10 for the greedy coloring algorithm. The number colors needed depends on the ordering of the vertices. If color the vertex with the highest degree first, on average, you can better results.

Sometimes $d + 1$ colors is the best you can do, e.g, K_n. It gives the best possible bound for any graph with degree bounded by d that has K_{d+1} as a subgraph. But sometimes $d + 1$ colors is far from the best that you can do.

9.4 Connectivity

9.4.1 Connected Components

DEFINITION 9.29 **Connected** Two vertices are connected in a graph when there is a path that begins at one and ends at the other. By convention, every vertex is connected to itself by a path of length zero.

DEFINITION 9.30 **Connected Graph** A graph is connected when every pair of vertices are connected.

DEFINITION 9.31 **Connected Component** The connected components of a graph are the equivalence classes of the vertices under the "is reachable from" relation.

LEMMA 9.32 A graph is connected iff it has exactly one connected component.

THEOREM 9.33 Every graph G has at least $|V| - |E|$ connected components.

Proof. We use proof on the number of vertices e. Let the inductive hypothesis by $P(e) :=$ every graph G with has at least $|V| - e$ connected components.
Base case: $e = 0$. In a graph with 0 edges, each vertex is itself a connected component, and so there are exactly $|V| = |V| - 0$ connected components.
Inductive step: Assume that $P(e)$ is true, then for a graph $G = (V, E)$ with $|E| = e + 1$, we remove an arbitrary edge $\{u, v\}$ and call the resulting graph G'. By the induction assumption, G' has at least $|V| - e$ connected components.

Now add back the edge $\{u, v\}$. There are two cases:
Case 1: If u and v were in the same connected components of G', then G has the same connected components as G', so G has at least $|V| - e > |V| - (e + 1)$ components.
Case 2: If u and v were in different connected components of G', then these two components are merged into one component in G, but all other components remain unchanged, reducing the number of components by 1. Therefore, G has at least $|V| - e - 1 = |V| - (e + 1)$ connected components.

So in either case, $P(e + 1)$ holds, which completes the inductive step. The theorem now follows by induction. □

THEOREM 9.34 Every connected graph with n vertices has at least $n - 1$ edges.

9.4.2 k-Connected Graphs

Connectivity measures fault tolerance of a network: how many connections can fail without cutting off communication.

DEFINITION 9.35 **k-Edge Connected Vertices** Two vertices in a graph are k-edge connected when they remain connected in every subgraph obtained by deleting up to $k - 1$ edges.

DEFINITION 9.36 **k-Edge Connected Graphs** A graph is k-edge connected when it has more than one vertex, and pair of distinct vertices in the graph are k-connected.

DEFINITION 9.37 **Cut Edge** If two vertices are connected in a graph G, but not connected when an edge e is removed, then e is called a cut edge of G.

A graph with more than one vertex is 2-connected iff it is connected and has no cut edges.

LEMMA 9.38 An edge is a cut edge iff it is not on a cycle.

THEOREM 9.39 **Menger's Theorem** If two vertices are k-edge connected, then there are k non-overlapping paths connecting them.

9.4.3 Summary

See Tab. 9.1.

9.5 Euler Tours and Hamiltonian Cycles

9.5.1 Euler Tours

DEFINITION 9.40 **Euler Walk** A walk that traverses every edge in a graph exactly once.

Table 9.1
Summary of graphs and its connectivities

Graphs	Connectivity	Number Edges
K_n	$n-1$	$n^2/2$
Grid	4	$2n$
Cycle	2	n
Tree	1	$n-1$

DEFINITION 9.41 **Euler tour** An Euler walk that starts and finishes at the same vertex.

THEOREM 9.42 A connected graph has an Euler tour if and only if every vertex has even degree.

COROLLARY 9.43 A connected graph has an Euler walk if and only if every precisely 0 or 2 nodes in G have odd degree.

9.5.2 Hamiltonian Cycles

DEFINITION 9.44 **Hamiltonian Cycle** A Hamiltonian cycle in a graph G is a cycle that visits every node in G exactly once.

Proof. $\Rightarrow$ **Part**: Assume that $G = (V, E)$ has a Euler tour $v_0, v_1, \ldots, v_k$, where $v_0 = v_k$. Since every edge is traversed once in the tour, $k = |E|$ and the degree of a node v is the number appearances in the sequence times two. Hence the degree of every node is even.
$\Leftarrow$ **Part**: Assume $G = (V, E)$ is a graph where every node has even degrees. Let $p := v_0, v_1, \ldots, v_k$ be the longest walk in G that traverses no edge more than once. p must traverse every edge incident to v_k; otherwise the walk could be extended and p would not be the longest walk that traverses all edges at most once. Moreover, it must be that $v_k = v_0$ since otherwise v_k could have odd degree in p. which is not possible by assumption.

We use proof by contradiction. Assume that p is not a Euler tour, we can find an edge not in p but incident to some vertex in p, call this edge $\{u, v_i\}$. But then we construct a walk p that is longer than p but that still uses no edge more than once: $u, v_i, v_{i+1}, \ldots, v_k, v_1, v_2, \ldots v_i$. This is a contradiction. □

DEFINITION 9.45 **Hamiltonian Path** A Hamiltonian path is a path in G that visits every node exactly once.

Determining whether a graph has a Hamiltonian cycle is the same category of problem as the SAT problem and the coloring problem.

9.5.3 The Traveling Salesperson Problem

DEFINITION 9.46 **Weight of a Cycle** Given a weighted graph G, the weight of a cycle in G is defined as the sum of the weights of the edges in the cycle.

10 Bipartite Graphs

10.1 Bipartite Graphs

DEFINITION 10.1 **Bipartite Graph** A bipartite graph is a graph $G = (V, E)$ whose vertices can be partitioned into two sets V_L and V_R, such that every edge has one endpoint in V_L and the other endpoint in V_R.

THEOREM 10.2 A graph G with at least one edge is bipartite iff $\chi(G) = 2$.

10.2 The Bipartite Matching Problem

10.2.1 The Matching Condition

DEFINITION 10.3 **Matching** Given a graph $G = (V, E)$, a matching is a set of edges of G such that no vertex is an endpoint of more than one edge in the matching. That means, the matching is a total injective function from V_L to V_R.

DEFINITION 10.4 **Cover** A matching is said to cover a set S of vertices iff each vertex in S is an endpoint of an edge of the matching.

DEFINITION 10.5 **Perfect Matching** A matching is said to be perfect if it covers V.

DEFINITION 10.6 **Neighbors** In any graph G, the set $N(S)$ of neighbors of some set S of vertices is the image of S under the edge-relation, that is,

$$N(S) = \{r \mid \exists s \in S . s - r \in E\}. \tag{10.1}$$

DEFINITION 10.7 **Bottleneck** S is called a bottleneck if

$$|S| > |N(S)|. \tag{10.2}$$

THEOREM 10.8 **Hall's Theorem** Let G be a bipartite graph. There is a matching in G that covers V_L iff no subset of V_L is a bottleneck.

Proof. First, suppose that a matching exists and show that the matching condition holds. Consider an arbitrary subset of V_L. Each vertex from V_L has an edge to V_R it is matched to. Therefore, every subset of $S \subseteq V_L$ has $|S| \leq |N(S)|$.

Next, suppose that the matching condition holds and show that a matching exists. We use proof by induction on $|V_L|$. Let the inductive hypothesis be $P(m) :=$ for any set of $|V_L| = m$, if the matching condition holds for V_L, then there is a matching for V_L.
Base case: $|V_L| = 1$, then the matching condition implies that only $v \in V_L$ has an edge to at least one $u \in V_R$, and so a matching exists.
Inductive step: Suppose that $P(m)$ is true, for $|V_L| = m + 1 \geq 2$, there are two cases

Case 1: Every proper subset $S \subset V_L$ has $|S| < |N(S)|$. Then we can pair an arbitrary $v \in V_L$ with $u \in V_R$ which $\{v, u\} \in E$ and remove both of them. The matching condition still holds for the remaining vertices, so we can match the rest of them by induction.
Case 2: There exists some proper subset $S \subset V_L$ which $|S| = |N(S)|$. We match S and $N(S)$ by induction and remove them. Consider an arbitrary subset of the remaining left vertices $S' \subset V_L - S$, and $N'(S')$ be the set of remaining right vertices they are adjacent to. Originally, the combined set $S \cap S'$ is adjacent to $N(S) \cap N'(S')$. So by the matching condition,

$$|S \cap S'| \leq |N(S) \cap N'(S')|\,. \tag{10.3}$$

After we remove S and $N(S)$, it must be $|S'| \leq |N'(S')|$. That is to say, the matching condition holds for the remaining vertices, and we can also math the rest of left vertices.

So in both cases, there is a matching for V_L, which completes the proof of the inductive step. The theorem follows by induction. □

10.2.2 An Easy Matching Condition

The bipartite matching condition requires that every subset of men has a certain property. However, there is a simple property of vertex degrees in a bipartite graph that guarantees the existence of a matching.

DEFINITION 10.9 **Degree-Constrained** A bipartite graph is degree-constrained if vertex degrees on the left are at least as large as those on the right. More precisely,

$$\forall l \in V_L, \forall r \in V_R. \deg l \geq \deg r\,. \tag{10.4}$$

THEOREM 10.10 If G is a degree-constrained bipartite graph, then there is a matching that covers V_L.

Proof. We use proof by contradiction. Suppose that G is a degree-constrained bipartite graph but that there is no matching that covers V_L. This means that there must be a bottleneck $S \subseteq V_L$.

Let x be the value such that

$$\forall l \in V_L, \forall r \in V_R. \deg l \geq x \geq \deg r\,. \tag{10.5}$$

Therefore,

$$\sum_{u \in S} \deg u \geq \sum_{u \in S} x = |S|x\,, \tag{10.6}$$

$$\sum_{v \in S} \deg v \leq \sum_{v \in S} x = |N(S)|x\,. \tag{10.7}$$

Since every edge incident to a node in S is incident to a node in $N(S)$, we know that

$$\sum_{u \in S} \deg u = \sum_{v \in S} \deg v \Rightarrow |S|x \leq |N(S)|x \Rightarrow |S| \leq |N(S)|. \tag{10.8}$$

This contradicts to the assumption that S is a bottleneck. □

DEFINITION 10.11 **Regular Graph** A graph is said to be regular if every node has the same degree.

THEOREM 10.12 Every regular bipartite graph has a perfect matching.

Proof. Since regular graphs are degree-constrained, there must be a matching that covers V_L. Since G is regular, we also know that $|L| = |R|$ and thus the matching must also covers V_R. This means that every node in G is incident to an edge in the matching and thus G has a perfect matching. □

10.3 The Stable Marriage Problem

10.3.1 The Stable Marriage Problem

DEFINITION 10.13 **Rogue Couple** Given a matching, a man and woman who are not married to each other and who like each other better than their spouses form a rogue couple.

DEFINITION 10.14 **Stable Matching** A stable matching is a matching with no rogue couples.

DEFINITION 10.15 **The Stable Marriage Problem** Given n men and n women, each person has preferences of the opposite-gender person they would like to marry: each man has his preference list of all the women, and each woman has her preference list of all of the men. The preferences don't have to be symmetric. The goal is to find a perfect matching which is also stable.

In fact, if people are supposed to be paired off as buddies, regardless of gender, a stable matching may not be possible. But when men are only allowed to marry women, and vice-versa, then there is always a stable matching among a group of men and women.

10.3.2 The Matching Algorithm

See Alg. 11

10.3.2.1 Termination

THEOREM 10.16 The Matching Algorithm will terminate in $\leq n^2 + 1$ days.

Algorithm 11 The Matching Algorithm.

Input: n men and n women, each person has preferences of the opposite-gender person they would like to marry.

Output: A perfect matching which is also stable.

The following events happen each day:

- Morning: Each man serenades the top choice among the women on his list. If a man has no women left on his list, he stays home.
- Afternoon: Each woman reject all but her favorite among them.
- Evening: Any man who is rejected by a woman crosses that woman off his preference list.

Termination condition: When a day arrives in which every woman has at most one suitor, the ritual ends with each woman marrying her suitor, if she has one.

Proof. Every day on which the ritual has not terminated, there must be some woman serenaded by at least two men, and at least one of them will have to cross her off his list.

Each of the n men's lists initially has n women on it, for a total of n^2 list entries. Since no women ever gets added to a list, the total number of entries on the lists is $\mathbb{N}$-valued and strictly decreasing, and so the ritual can continue for $\leq n^2$ days. □

10.3.2.2 Partial Correctness

LEMMA 10.17 The following is a preserved invariant of the Matching Algorithm: $P :=$ for every woman w and man m, if w is crossed off in m's list, then w has a suitor whom she prefers over m.

Proof. w is crossed off in m's list only when w has a suitor she prefers to m. Thereafter, her favorite suitor does not change until one she likes better comes along. So if her favorite suitor was preferable to m, then any new favorite suitor will be as well. □

LEMMA 10.18 The rank of a girl's favorite is weakly increasing.

LEMMA 10.19 The rank of girl a boy serenades is weakly decreasing.

THEOREM 10.20 Everyone is married at the end of the Mating Ritual.

Proof. We use proof by contradiction. Assume that on the last day of the Mating Ritual, there exists a man m not married. This means m cannot be serenading anybody, that is, his list must be empty. So every woman must have been crossed off his list and since P is true, every woman has a suitor whom she prefers to m.

In particular, every woman has some suitor, and since it is the last day, they have only one suitor, and this is who they marry. But there are an equal number of men and women,

and so if all women are married, so are all men, contradicting the assumption that m is not married. □

THEOREM 10.21 The Matching Algorithm produces a stable matching.

Proof. Let m and w be any man and woman, respectively, that are not married to each other on the last day of the Matching Ritual. We will prove that m and w are not a rouge couple, and thus that all marriages on the last day are stable. There are two cases.
Case 1: w is not on m's list by the end. Then by preserved invariant P, w has a suitor (and hence a husband) whom she prefers to m. So w and m cannot be a rogue couple.
Case 2: w is on m's list. Since m picks women to serenade by working down his list, his wife must be higher on his preference list than w. So w and m cannot be a rogue couple. □

10.3.2.3 Fairness

DEFINITION 10.22 **Feasible Spouse** Given a set of preferences for the men and women, one person is a feasible spouse for another person when there is a stable matching in which these two people are married.

LEMMA 10.23 The following is a preserved invariant: $Q :=$ for every woman w and man m, if w is crossed off in m's list, then w is not a feasible spouse for m.

Proof. Suppose Q holds at some point in the Ritual and some woman w is about to be crossed off some man w's list. When w gets crossed of m's list, it is because w has a suitor m' she prefers to m. Since Q holds, all m''s feasible wives and still on his list, and w is at the top. So m' likes w better than all his other feasible spouses.

Now if w could be married to m in some set of stable marriage, then m' must be married to a wife he likes less than w, making w and m' a rogue couple and contradicting stability. So w cannot be married to m, that is, w is not a feasible wife for m, as claimed. □

DEFINITION 10.24 **Optimal Spouse** A person's optimal spouse is their most preferred feasible spouse.

DEFINITION 10.25 **Pessimal Spouse** A person's pessimal spouse is their least preferred feasible spouse.

THEOREM 10.26 The Matching Algorithm marries every man to his optimal spouse and every woman to her pessimal spouse.

Proof. If m is married to w on the final day of the Ritual, then everyone above w on m's preference list was crossed off, and by property Q, all these crossed off women were infeasible for m. So w is m's highest ranked feasible spouse, that is, his optimal spouse.

Further, since m likes w better than any other feasible wife, m and w would be a rogue couple if w was married to a husband m' she likes less than m. So m must be w's least feasible husband. □

11 Planar Graphs

11.1 Definitions of Planar Graphs

11.1.1 A Planar Geometry Definition for Planar Graphs

DEFINITION 11.1 **Drawing** A drawing of a graph assigns to each node a distinct point in the plane and assigns to each edge a smooth curve in the plane whose endpoints correspond to the nodes incident to the edge.

DEFINITION 11.2 **Planar Drawing** The drawing is planar if none of the curves cross themselves or other curves, namely, the only points that appear more than once on any of the curves are the node points.

DEFINITION 11.3 **Planar Graphs** A graph is planar when it has a planar drawing.

11.1.2 A Recursive Definition for Planar Graphs

DEFINITION 11.4 **Continuous Faces** The curves in a planar drawing divide up the plane into connected regions called the continuous faces of the drawing.

DEFINITION 11.5 **Outside Face** The face extends off to infinity in all directions.

DEFINITION 11.6 **Discrete Faces** The vertices along the boundary of each continuous face form a closed walk. Thus, we can identify each of the faces by its closed walk. Those closed walk are called the discrete faces.

Apart from those 2 exceptions, the closed walks in discrete faces are also cycles.

- If the graph has a bridge (cut edge), the sequence of vertices along the boundary of the outer region defines a closed walk, but not a cycle, since the walk has two occurrences of the bridge and each of its endpoints.
- If the graph has a dongle, the sequence of vertices along the boundary of the inner region defines a closed walk, but not a cycle, since the walk has two occurrences of every dongle — once "coming" and once "going".

DEFINITION 11.7 **Bridge** The edges that occur twice on the same discrete face.

DEFINITION 11.8 **Dongle** Tree made of bridges.

DEFINITION 11.9 **Planar Embedding** A planar embedding of a connected graph consists of a nonempty set of closed walks of the graph called the discrete faces of the embedding. Planar embeddings are defined recursively as follows:

- Base case: If G is a graph consisting of a single vertex v, then a planar embedding of G has one discrete face, namely the length zero closed walk v.

- Constructor case (split a face): Suppose G is a connected graph with a planar embedding, and suppose u and v are distinct, nonadjacent vertices of G that occur in some discrete face γ of the planar embedding.[1] That is, γ is a closed walk of the form

$$\gamma = (\alpha, \beta), \tag{11.1}$$

  where α is a walk from u to v and β is a walk from v to u. Then the graph obtained by adding the edge $u - v$ to the edges of G has a planar embedding with the same discrete faces as G, except that face γ is replaced by the two discrete faces

$$(\alpha, v - u) \quad \text{and} \quad (u - v, \beta). \tag{11.2}$$

- Constructor case (add a bridge): Suppose G and H are connected graphs with planar embeddings and disjoint sets of vertices. Let γ be a discrete face of the embedding of G and suppose that γ begins and ends at vertex u. Similarly, let δ be a discrete face of the embedding of H that begins and ends at vertex v. Then the graph obtained by connecting G and H with a new edge uv,[2] has a planar embedding whose discrete faces are the union of the discrete faces of G and H, except that faces γ and *delta* are replaced by one new face

$$(\gamma, a - b, \delta, b - a). \tag{11.3}$$

In general, a graph is planar because it has a planar drawing iff each of its connected components has a planar embedding.

Every planar drawing has an immediately-recognizable outer face. In fact, a planar embedding could be drawn with any given face on the outside. So pictures that show different "outside" boundaries may actually be illustrations of the same planar embedding.

11.2 Bounding the Number of Edges

11.2.1 Euler's Formula

THEOREM 11.10 **Euler's Formula** If a connected graph has a planar embedding, then # vertices and edges completely determines # faces in every possible planar embedding of the graph, i.e.,

$$|V| - |E| + |F| = 2. \tag{11.4}$$

1 A new curve won't cross any other curves precisely when it stays within one of the continuous faces.

2 A new curve won't have to cross any other curves if it can go between the outer faces of two different drawings.

11.2.2 Bounding the Number of Edges of Connected Graphs

LEMMA 11.11 In a planar embedding of a connected graph, each edge occurs once in each of two different faces, or occurs exactly twice in one face.

LEMMA 11.12 In a planar embedding of a connected graph with at least three vertices, each face is of length at least 3.

THEOREM 11.13 Suppose a connected planar graph has $|V| \geq 3$ vertices and $|E|$ edges, then

$$|E| \leq 3|V| - 6\,. \tag{11.5}$$

COROLLARY 11.14 K_5 is not planar.

11.2.3 Bounding the Number of Edges of Connected Bipartite Graphs

LEMMA 11.15 In a planar embedding of a connected bipartite graph with at least three vertices, each face is of length at least 4.

THEOREM 11.16 Suppose a connected planar graph has $|V| \geq 3$ vertices and $|E|$ edges, then

$$|E| \leq 2|V| - 4\,. \tag{11.6}$$

DEFINITION 11.17 **Complete Bipartite Graph** $K_{n,n}$ A bipartite graph G with $|V_L| = |V_R| = n$, and there is a edge between each node in V_L and each node in V_R.

COROLLARY 11.18 $K_{3,3}$ is not planar.

11.2.4 Kuratowski's Theorem

DEFINITION 11.19 **Minor** A minor of a graph G is a graph that can be obtained by repeatedly deleting vertices, deleting edges, and merging adjacent vertices of G.

THEOREM 11.20 **Kuratowski** A graph is not planar if and only if it contains K_5 or $K_{3,3}$ as a minor.

11.3 Coloring Planar Graphs

LEMMA 11.21 Any subgraph of a planar graph is planar.

LEMMA 11.22 Merging two adjacent vertices of a planar graph leaves another planar graph.

LEMMA 11.23 For every planar graph, there exists a vertex of degree at most 5.

THEOREM 11.24 Every planar graph is 5-colorable.

11.4 Classifying Polyhedra

DEFINITION 11.25 **Polyhedron** A polyhedron is a convex, three-dimensional region bounded by a finite number of polygonal faces.

DEFINITION 11.26 **Regular** If the faces are identical regular polygons and an equal number of polygons meet at each corner, then the polyhedron is regular.

Suppose we took any polyhedron and placed a sphere inside it. Then we could project the polyhedron face boundaries onto the sphere, which would give an image that was a planar graph embedded on the sphere, with the images of the corners of the polyhedron corresponding to vertices of the graph.

THEOREM 11.27 Let m be the number of faces that meet at each corner of a polyhedron, and let n be the number of edges on each face. In the corresponding planar graph, there are m edges incident to each of the $|V|$ vertices. Then we have

$$\frac{1}{m} + \frac{1}{n} = \frac{1}{|E|} + \frac{1}{2}. \tag{11.7}$$

This equation only has 5 solutions.

12 Forests and Trees

12.1 Rooted and Ordered Trees

12.1.1 Rooted and Ordered Trees

DEFINITION 12.1 **Forest** An acyclic graph.

DEFINITION 12.2 **Tree** A connected acyclic graph. Each of the forest's connected components is a tree.

DEFINITION 12.3 **Rooted Tree** T A tree in which one of the vertices is distinguished from the others. We call the distinguished vertex the **root** of the tree. We often refer to a vertex of a rooted tree as a **node** of the tree.

DEFINITION 12.4 **Ancestor** For a node x in a rooted tree T with root r. We call any node y on the unique simple path from r to x an ancestor of x. If $y \neq x$, then y is a **proper ancestor** of x.

DEFINITION 12.5 **Descendant** If y is an ancestor of x, then x is a descendant of y. If $y \neq x$, then x is a **proper descendant** of y.

DEFINITION 12.6 **Subtree** The subtree rooted at x is the tree induced by descendants of x, rooted at x.

DEFINITION 12.7 **Parent** If the last edge on the simple path from the root r of a tree T to a node x is $\{y,x\}$, then y is the parent of x, and x is a **child** of y.

DEFINITION 12.8 **Siblings** Nodes have the same parent.

DEFINITION 12.9 **Leaf/External Node** A node with no children.

DEFINITION 12.10 **Internal Node** A nonleaf node.

DEFINITION 12.11 **Degree** The number of children of a node x in a rooted tree T is the degree of x.

DEFINITION 12.12 **Depth** The length of the path from the root r to a node x is the depth of x in T.

DEFINITION 12.13 **Level** All nodes at the same depth.

DEFINITION 12.14 **Height** The height of a node in a tree is the number of edges on the longest downward path from the node to a leaf. The height of a tree is the height of its root, which is also equal to the largest depth of any node in the tree.

DEFINITION 12.15 **Ordered Tree** A rooted tree in which the children of each node are ordered.

12.1.2 Properties

THEOREM 12.16 Every connected subgraph of a tree is also a tree.

Proof. We proof the contrapositve. Assume that there exists a connected subgraph which contains a cycle, then that cycle is also a cycle in the whole graph. □

THEOREM 12.17 There is a unique path between every pair of vertices.

Proof. Since a tree is connected, there is at least one path between every pair of vertices. Suppose for contradiction that there are two different paths between some pair of vertices x and y. Beginning at x, let u be the first vertex where the paths diverge, and v be the next vertex they share. Then there are two paths from u to v with no common edges, which defines a cycle. This is a contradiction. □

THEOREM 12.18 Adding an edge between nonadjacent nodes in a tree creates a graph with a cycle.

Proof. An additional edge $\{x, y\}$ together with the unique path between x and y forms a cycle. □

THEOREM 12.19 Removing any edge disconnects the graph. That is, every edge is a cut edge.

Proof. Suppose we remove $\{x, y\}$, since the tree contained a unique path between x and y, that path must have been $\{x, y\}$. Therefore, when that edge is removed, no path remains, and so the graph is not connected. □

THEOREM 12.20 If the tree has at least 2 vertices, then it has at least 2 leaves.

Proof. For a tree T with ≥ 2 vertices, let $v_1, v_2, \ldots, v_k$ be the sequence of vertices on a longest path in the tree. Then $k \geq 2$ since a tree with two vertices must contain at least one edge.

$\forall i \in (2, k]$, there cannot be an edge $\{v_1, v_i\}$; otherwise, vertices $v_1, v_2, \ldots, v_i$ form a cycle. There cannot be an edge $\{u, v_1\}$ where u is not on the path; otherwise, we could make the path longer. Therefore, the only edge incident to v_1 is $\{v_1, v_2\}$, which means that v_1 is a leaf. By a symmetric argument, v_m is a second leaf. □

THEOREM 12.21 A graph G is a tree iff G is a forest and $|V| = |E| + 1$.

Proof. We use proof by induction. Let the inductive hypothesis $P(n) :=$ there are $n - 1$ edges in any n-vertex tree.

Base case: $P(1)$ is true since a tree with 1 node has 0 edges.
Inductive step: Assume $P(n)$, consider an $(n+1)$-vertex tree T. Let v be a leaf of the tree. You can verify that deleting a vertex of degree 1 (and incident edge) from any connected graph leaves a connected subgraph. So deleting v and its incident edge gives a smaller tree T', and T' has $n-1$ edges by induction. If we re-attach the vertex v and its incident edge, then we find that T has $n = (n+1) - 1$ edges. Hence $P(n+1)$ is true, and the induction proof is complete. □

12.2 Binary and Positional Trees

DEFINITION 12.22 **Binary Tree** A binary tree T is a structure defined on a finite set of nodes that either

- contains no nodes, or
- is composed of three disjoint set of nodes: a root node, a binary tree called its **left subtree**, and a binary tree called its **right subtree**.

DEFINITION 12.23 **Empty Tree/Null Tree** The binary tree contains no nodes.

DEFINITION 12.24 **Absent/Missing** When a subtree is the null tree, we say that the child is absent or missing.

DEFINITION 12.25 **Full Binary Tree** Each node is either a leaf or has degree exactly 2. There is no degree-1 nodes.

DEFINITION 12.26 **Positional Tree** The children of a node are labeled with distinct positive integers. The i-th child of a node is absent if no child is labeled with integer i.

DEFINITION 12.27 **k-Ary Tree** A positional tree in which for every node, all children with labels greater than k are missing. Thus, a binary tree is a k-ary tree with $k = 2$.

DEFINITION 12.28 **Complete k-Ary Tree** A k-ary tree in which all leaves have the same depth and all internal nodes have degree k.

LEMMA 12.29 The height of a complete k-ary tree with n leaves is $\log_k n$.

Proof. The number of leaves at depth i is k^i. □

THEOREM 12.30 The number of internal nodes of a complete k-ary tree with height h is $\frac{k^h-1}{k-1}$.

Proof. We add up each level

$$1 + k + k^2 + \cdots + k^{h-1} = \sum_{i=0}^{h-1} k^i = \frac{k^h - 1}{k-1}. \tag{12.1}$$

□

COROLLARY 12.31 The number of internal nodes of a complete binary tree with height h is $2^h - 1$.

Proof. Substitute $k = 2$. □

12.3 Spanning Trees

12.3.1 Spanning Subgraphs and Spanning Trees

DEFINITION 12.32 **Spanning Subgraph** A subgraph containing all the vertices of the graph G.

DEFINITION 12.33 **Spanning Tree** A spanning subgraph that is a tree.

THEOREM 12.34 Every connected graph contains a spanning tree.

Proof. We use proof by contradiction. Assume there is some connected graph G that has no spanning tree, and let T be a connected subgraph of G, with the same vertices as G, and with the smallest number of edges possible for such a subgraph. By the assumption, T is not a spanning tree and so it contains some cycle: $v_0, v_1, v_2, \ldots, v_k, v_0$.

Suppose that we remove the last edge $\{v_k, v_0\}$. If a pair of vertices x and y was joined by a path not containing $\{v_k, v_0\}$, then they remain joined by that path. On the other hand, if x and y was joined by a path containing $\{v_k, v_0\}$, they remain joined by a walk containing the remainder of the cycle, then they must be joined by a path. So all the vertices are still connected after we remove an edge from T. This contradicts to the assumption that T is the minimum size connected subgraph with all the vertices of G. □

12.3.2 Minimum Weight Spanning Trees

DEFINITION 12.35 **Weight of a Graph** Sum of the weights of its edges.

DEFINITION 12.36 **Minimum Weight Spanning Tree (MST)** A minimum weight spanning tree of an edge-weighted graph G is a spanning tree of G with the smallest possible sum of edge weights.

12.3.3 Find a MST

The standard methods for finding MST's all start with the empty spanning forest and build up to an MST by adding one extending edge after another. See Alg. 12, 13. Both algorithms work, and we will only analysis the Kruskal's MST Algorithm for simplicity.

Algorithm 12 Prim's MST Algorithm.

Input: A connected graph G.
Output: MST of G.

Grow a *tree* one edge at a time by adding a minimum weight edge possible to the tree.

Algorithm 13 Kruskal's MST Algorithm.

Input: A connected graph G.
Output: MST of G.

Grow a *forest* one edge at a time by adding a minimum weight edge possible to the forest.

LEMMA 12.37 For any $k \in \mathbb{N}$, let S consist of the fist k edges selected by Prim's MST Algorithm. Then there exists some MST $T = (V, E)$ for G such that $S \subseteq E$, that is, the set of edges that we are growing is always contained in some MST.

Proof. We use proof by induction. Let the inductive hypothesis $P(m) :=$ for any G and any set S of m edges initially selected, there exists an MST $T = (V, E)$ of G such that $S \subseteq E$.
Base case: $m = 0$, then $S = \varnothing$, so $S \subseteq E$ trivially holds for any MST T.
Inductive step: Assume $P(m)$, let e denote the $(m+1)$-st edge selected by the Prim's MST Algorithm, and let S denote the first m edges selected, and let $T = (V, E)$ be the MST such that $S \subseteq E$, which exists by inductive hypothesis. There are two cases.
Case 1: $e \in E$, then $S \cup \{e\} \subseteq E$, and thus $P(m+1)$ holds.
Case 2: $e \notin E$, then since T is a tree, adding e to T will form a cycle. Moreover, the cycle cannot only contain edges in S, since e was chosen so that together with S, it does not form a cycle.

This implies that $E \cup \{e\}$ contains a cycle that contains a edge $e' \in E - S$. Note that the weight of e is at most e'. since Kruskal's MST Algorithm picks the minimum weight edge that does not make a cycle in S. Let $T^* = (V, E^*)$ where $E^* :=(E - \{e'\}) \cup \{e\}$, that is, we swap e and e' in T.

We claim that T^* is a MST. T^* is acyclic since it was produced by removing and edge form the only cycle in $E \cup \{e\}$. T^* is connected since the edge we deleted form $E \cup \{e\}$ was on a cycle. Since T^* contains all the nodes of G, it must be a spanning tree for G. Therefore, since the the weight of e is at most e', the weight of T^* is at most that of T, and thus T^* is a MST for G.

Since $S \cup \{e\} \subseteq E^*$. $P(m+1)$ holds, and $P(m)$ holds for all m. □

THEOREM 12.38 For any connected, weighted graph G, Kruskal's MST Algorithm produces an MST for G.

Proof. As long as there are fewer than $n-1$ edges picked, there exists some edge in $E-S$ and so there is an edge that we can add to S without forming a cycle. Once $k=n-1$, we know that S is an MST. □

III COUNTING

13 Asymptotics and Summations

13.1 Asymptotic Notation

Asymptotic notation is a shorthand used to give a quick measure of the behavior of a function $f(n)$ as n grows large.

13.1.1 Little Oh Notation

Little oh indicates that one function grows at a significantly slower rate than another.

DEFINITION 13.1 **Asymptotically Smaller** o For functions $f, g \ \mathbb{R} \mapsto \mathbb{R}$, with g nonnegative, we say f is asymptotically smaller than g, in symbols, $f(n) = o(g(n))$, iff

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0\,. \tag{13.1}$$

LEMMA 13.2 Little oh is a strict partial order.

Proof. **Irreflexive**

$$\lim_{n\to\infty} \frac{f(n)}{f(n)} = 1 \neq 0\,. \tag{13.2}$$

Therefore $f(n) \neq o(f(n))$.

Transitive. If $f(n) = o(g(n))$ and $g(n) = o(h(n))$, then

$$\lim_{n\to\infty} \frac{f(n)}{h(n)} = \lim_{n\to\infty} \frac{f(n)}{g(n)} \frac{g(n)}{h(n)} = 0\,. \tag{13.3}$$

Therefore $f(n) \neq o(h(n))$. □

LEMMA 13.3

$$\forall b > a \geq 0.\ n^a = o(n^b)\,. \tag{13.4}$$

Proof.

$$\lim_{n\to\infty} \frac{n^a}{n^b} = \lim_{n\to\infty} \frac{1}{n^{b-a}} = 0\,. \tag{13.5}$$

□

COROLLARY 13.4

$$\forall c > 0.\ \log n = o(n^c)\,. \tag{13.6}$$

Proof. [1] Using the familiar fact that

$$\forall x > 1.\ \log x < x\,. \tag{13.7}$$

Let $x = n^d$, where $c > d > 0$. Therefore $x = n^d > 1$, then by Lemma 13.3,

$$\log n^d = d \log n < n^d = o(n^c)\,. \tag{13.8}$$

That is to say,

$$\lim_{n\to\infty} \frac{d \log n}{n^c} = d \lim_{n\to\infty} \frac{\log n}{n^c} = 0\,. \tag{13.9}$$

$$\therefore \lim_{n\to\infty} \frac{\log n}{n^c} = 0\,. \tag{13.10}$$

$$\therefore \log n = o(n^c)\,. \tag{13.11}$$

□

COROLLARY 13.5

$$\forall a, b \in \mathbb{R}, a > 1.\ n^b = o(a^n)\,. \tag{13.12}$$

Proof. By l'Hopital's Rule.

$$\lim_{n\to\infty} \frac{n^b}{a^n} = \lim_{n\to\infty} \frac{b!}{(\log a)^{b!} a^n} = 0\,. \tag{13.13}$$

□

13.1.2 Big Oh Notation

Big oh indicates that one function grows not much more rapidly than another.

DEFINITION 13.6 **Asymptotically Smaller or Equal** O For functions $f, g\ \mathbb{R} \mapsto \mathbb{R}$,, with g nonnegative, we say f is asymptotically equal to g, in symbols, $f(n) = O(g(n))$, iff

$$\lim_{n\to\infty} \frac{|f(n)|}{g(n)} < \infty\,. \tag{13.14}$$

1 It can also be proved by l'Hopital's Rule.

LEMMA 13.7

$$f(n) \sim g(n) \Rightarrow f(n) = O(g(n)), \tag{13.15}$$

$$f(n) = o(g) \Rightarrow f(n) = O(g(n)). \tag{13.16}$$

Beside, the converse is not true.

Proof.

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 1 < \infty, \tag{13.17}$$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0 < \infty. \tag{13.18}$$

We prove the converse is not true by giving a counter example. For example, $2x = O(x)$, but $2x \not\sim x$ and $2x \neq o(x)$. □

An equivalent, more usual formulation of big Oh notation is as follows,

DEFINITION 13.8 **Asymptotically Smaller or Equal** O For functions $f, g\ \mathbb{R} \mapsto \mathbb{R}$,, with g nonnegative, we say f is asymptotically smaller than or equal to g, in symbols, $f(n) = O(g(n))$, iff

$$\exists c \geq 0, n_0, \forall n \geq n_0.\ |f(n)| \leq cg(n). \tag{13.19}$$

It means $f(n)$ is less than or equal to $g(n)$, except that we are willing to ignore a constant factor c, and to allow exceptions for small n, namely, $n < n_0$.

LEMMA 13.9

$$\sum_{i=0}^{k} a_i x^i = O(x^k). \tag{13.20}$$

Proof.

$$\lim_{n\to\infty} \frac{\sum_{i=0}^{k} a_i x^i}{x^k} = a_k + \sum_{i=0}^{k-1} \lim_{n\to\infty} \frac{a_i}{x^{k-i}} = a_k < \infty. \tag{13.21}$$

□

13.1.3 Little Omega Notation

DEFINITION 13.10 **Asymptotically Bigger** ω For functions $f, g\ \mathbb{R} \mapsto \mathbb{R}$,, with g nonnegative, we say f is asymptotically bigger than g, in symbols, $f(n) = \omega(g(n))$, iff

$$\lim_{n\to\infty} \frac{|f(n)|}{g(n)} = \infty. \tag{13.22}$$

13.1.4 Big Omega Notation

DEFINITION 13.11 **Asymptotically Bigger or Equal** For functions $f, g \ \mathbb{R} \mapsto \mathbb{R},$, with g nonnegative, we say f is asymptotically bigger than or equal to g, in symbols, $f(n) = \quad (g(n))$, iff

$$\lim_{n\to\infty} \frac{|f(n)|}{g(n)} > 0\,. \tag{13.23}$$

13.1.5 Theta Notation

Theta indicates two functions are equal to within a constant factor.

DEFINITION 13.12 **Asymptotically Equal** Θ For functions $f, g \ \mathbb{R} \mapsto \mathbb{R},$, with g nonnegative, we say f is asymptotically equal to g, in symbols, $f(n) = O(g(n))$, iff

$$f(n) = O(g(n)) \wedge g(n) = O(f(n))\,. \tag{13.24}$$

That is to say,

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} \in (0,\infty)\ . \tag{13.25}$$

The Theta notation allows us to highlight growth rates and suppress distracting factors and low-order terms.

LEMMA 13.13 Theta is an equivalence relation.

Proof. **Reflexive**.

$$\lim_{n\to\infty} \frac{f(n)}{f(n)} = 1 \in (0,\infty)\ . \tag{13.26}$$

Therefore $f(n) = \Theta(f(n))$.

Symmetric. If $f(n) = \Theta(g(n))$, then

$$\lim_{n\to\infty} \frac{g(n)}{f(n)} = \frac{1}{\lim_{n\to\infty} \frac{f(n)}{g(n)}} \in (0,\infty)\ . \tag{13.27}$$

Therefore $g(n) = \Theta(g(n))$.

Transitivity. If $f(n) = \Theta(g(n))$ and $g(n) = \Theta(h(n))$, then

$$\lim_{n\to\infty} \frac{f(n)}{h(n)} = \lim_{n\to\infty} \frac{f(n)}{g(n)} \frac{g(n)}{h(n)} = \lim_{n\to\infty} \frac{f(n)}{g(n)} \lim_{n\to\infty} \frac{g(n)}{h(n)} \in (0,\infty)\ . \tag{13.28}$$

Therefore $f(n) = \Theta(h(n))$. □

13.1.6 Tilde Notation

Tilde notation indicates that two functions grow at the same rate.

DEFINITION 13.14 **Asymptotically Equal** $\sim$ For functions $f, g \ \mathbb{R} \mapsto \mathbb{R}$, we say f is asymptotically equal to g, in symbols, $f(n) \sim g(n)$, iff

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 1\,. \tag{13.29}$$

The reason that the tilde notation is useful is that often we do not care about lower order terms.

LEMMA 13.15 Tilde is an equivalence relation.

Proof. **Reflexive**.

$$\lim_{n\to\infty} \frac{f(n)}{f(n)} = 1\,. \tag{13.30}$$

Therefore $f(n) \sim f(n)$.

Symmetric. If $f(n) \sim g(n)$, then

$$\lim_{n\to\infty} \frac{g(n)}{f(n)} = \frac{1}{\lim_{n\to\infty} \frac{f(n)}{g(n)}} = 1\,. \tag{13.31}$$

Therefore $g(n) \sim f(n)$.

Transitivity. If $f(n) \sim g(n)$ and $g(n) \sim h(n)$, then

$$\lim_{n\to\infty} \frac{f(n)}{h(n)} = \lim_{n\to\infty} \frac{f(n)}{g(n)} \frac{g(n)}{h(n)} = \lim_{n\to\infty} \frac{f(n)}{g(n)} \lim_{n\to\infty} \frac{g(n)}{h(n)} = 1\,. \tag{13.32}$$

Therefore $f(n) \sim h(n)$. □

13.1.7 Summary

See Tab. 13.1.

13.2 Summation Formulas and Properties

13.2.1 Summation Formulas

DEFINITION 13.16 **Finite Summation** Given a sequence $(a_1, a_2, \ldots, a_n)$ of numbers where $n \in \mathbb{N}$,

$$\sum_{i=1}^{n} a_i := a_1 + a_2 + \cdots a_n\,. \tag{13.33}$$

Table 13.1
Summary of asymptotic notations.

Notation	Analogy	Meaning
$f(n) = o(g(n))$	$<$	$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$
$f(n) = O(g(n))$	$\leq$	$\lim_{n\to\infty} \frac{\lvert f(n)\rvert}{g(n)} < \infty$
$f(n) = \omega(g(n))$	$>$	$\lim_{n\to\infty} \frac{\lvert f(n)\rvert}{g(n)} = \infty$
$f(n) = \quad (g(n))$	$\geq$	$\lim_{n\to\infty} \frac{\lvert f(n)\rvert}{g(n)} > 0$
$f(n) = \Theta(g(n))$	roughly $=$	$\lim_{n\to\infty} \frac{\lvert f(n)\rvert}{g(n)} \in (0, \infty)$
$f(n) \sim g(n)$	nearly $=$	$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 1$

If $n = 0$, the value of summation is defined to be 0. The value of a finite series is always well defined, and we can add its terms in any order.

DEFINITION 13.17 **Infinite Summation** Given a sequence $(a_1, a_2, \ldots)$ of numbers,

$$\sum_{i=1}^{\infty} a_i := \lim_{n\to\infty} \sum_{i=1}^{n} a_i = a_1 + a_2 + \cdots . \tag{13.34}$$

If the limit does not exist, the series **diverge**, otherwise, it **converge**.

DEFINITION 13.18 **Absolutely Convergent Series** A sequence $(a_1, a_2, \ldots)$ of numbers is said to be an absolutely convergent series iff

$$\sum_{i=1}^{\infty} |a_i| \tag{13.35}$$

converges.

LEMMA 13.19 We can rearrange the terms of an absolutely convergent series.

13.2.2 Linearity

THEOREM 13.20 **Linearity of Summation** For any $c_1, c_2 \in \mathbb{R}$, and any finite sequences $(a_1, a_2, \ldots, a_n)$ and $(b_1, b_2, \ldots, b_n)$,

$$\sum_{i=1}^{n} (c_1 a_i + c_2 b_i) = c_1 \sum_{i=1}^{n} a_i + c_2 \sum_{i=1}^{n} b_i . \tag{13.36}$$

The linearity property also applies to infinite convergent series.

COROLLARY 13.21

$$\sum_{i=1}^{n} \Theta(f(i)) = \Theta\left(\sum_{i=1}^{n} f(i)\right). \tag{13.37}$$

The Theta notation on the left-hand side applies to the variable i, but on the right-hand side, it applies to n. This property also applies to infinite convergent series.

13.2.3 Telescoping Series

THEOREM 13.22 **Telescoping Series** For any sequence $(a_1, a_2, \ldots, a_n)$,

$$\sum_{i=1}^{n-1}(a_{i+1} - a_i) = a_n - a_1\,, \tag{13.38}$$

$$\sum_{i=1}^{n-1}(a_i - a_{i+1}) = a_1 - a_n\,. \tag{13.39}$$

Proof.

$$\sum_{i=1}^{n-1}(a_{i+1} - a_i) = \sum_{i=2}^{n} a_i - \sum_{i=1}^{n-1} a_i = a_n - a_1\,, \tag{13.40}$$

$$\sum_{i=1}^{n-1}(a_i - a_{i+1}) = \sum_{i=1}^{n-1} a_i - \sum_{i=2}^{n} a_i = a_1 - a_n\,. \tag{13.41}$$

□

COROLLARY 13.23

$$\sum_{i=1}^{n-1} \frac{1}{i(i+1)} = 1 - \frac{1}{n}\,. \tag{13.42}$$

Proof.

$$\sum_{i=1}^{n-1} \frac{1}{i(i+1)} = \sum_{i=1}^{n-1} \left(\frac{1}{i} - \frac{1}{i+1}\right) = 1 - \frac{1}{n}\,. \tag{13.43}$$

□

13.3 Arithmetic Sum

DEFINITION 13.24 **Closed Forms** Expressions that do not make use of subscripted summations or products, or those handy but sometimes troublesome sequences of three dots.

THEOREM 13.25 **Arithmetic Sum**

$$\forall n \in \mathbb{N}. \sum_{i=1}^{n} i = \frac{n(n+1)}{2} = \Theta(n^2). \tag{13.44}$$

Proof. Define

$$S = \sum_{i=1}^{n} i. \tag{13.45}$$

We compute S by using the perturbation method.

DEFINITION 13.26 **The Perturbation Method** Given a sum that has a nice structure, the perturbation method is to "perturb" the sum so that we can somehow combine the sum with the perturbation to get something much simpler.

The perturbation would be adding the sum to itself with the terms in reverse order

$$S = \sum_{i=1}^{n} (n+1-i). \tag{13.46}$$

$$\therefore\ 2S = \sum_{i=1}^{n} i + \sum_{i=1}^{n} (n+1-i) = \sum_{i=1}^{n} (n+1) = n(n+1). \tag{13.47}$$

$$\therefore\ S = \frac{n(n+1)}{2}. \tag{13.48}$$

□

The methods we develop for sums will also work for products, since any product can be converted into a sum by taking its logarithm.

13.4 Sums of Squares and Cubes

THEOREM 13.27

$$\forall n \in \mathbb{N}. \sum_{i=1}^{n} i^2 = \frac{(2n+1)(n+1)n}{6}. \tag{13.49}$$

Proof. The differentiating and integrating method does not work for summing consecutive squares. We use the **guess and proof** method.

We guess that the result might be a third-degree polynomial in n, since the sum contains n terms that average out to a value that grows quadratically in n. So we might guess that

$$\sum_{i=1}^{n} i^2 = an^3 + bn^2 + cn + d\,. \tag{13.50}$$

We can determine the parameters a, b, c and d by plugging in a few values for n.

$$n = 0 \Rightarrow 0 = d\,, \tag{13.51}$$

$$n = 1 \Rightarrow 1 = a + b + c + d\,, \tag{13.52}$$

$$n = 2 \Rightarrow 5 = 8a + 4b + 2c + d\,, \tag{13.53}$$

$$n = 3 \Rightarrow 14 = 27a + 9b + 3c + d\,. \tag{13.54}$$

Solving this system gives the solution

$$\sum_{i=1}^{n} i^2 = \frac{1}{3}n^3 + \frac{1}{2}n^2 + \frac{1}{6}n\,. \tag{13.55}$$

At last, we need to prove it by induction, which is omitted. □

THEOREM 13.28

$$\forall n \in \mathbb{N},\ \sum_{i=1}^{n} i^3 = \frac{n^2(n+1)^2}{4}\,. \tag{13.56}$$

13.5 Geometric Sum and Series

13.5.1 The Value of an Annuity

DEFINITION 13.29 ***n*-year, *m*-payment Annuity** A financial instrument that pays m dollars at the beginning of each year for n years.

Let's assume that money can be invested at a fixed annual interest rate p. For example, m dollars invested today will become $(1+p)m$ dollars in a year, $(1+p)^2 m$ dollars in two years, and $(1+p)^n m$ dollars in n years. Looked at another way, m dollars paid out a year from now is only really worth $\frac{m}{1+p}$ dollars today, m dollars paid out two years from now is only really worth $\frac{m}{(1+p)^2}$ dollars today, m dollars paid out n years from now is only really worth $\frac{m}{(1+p)^n}$ dollars today.

So for an n-year, m-payment annuity, the total total value V of the annuity is

$$V = \sum_{i=1}^{n} \frac{m}{(1+p)^{i-1}} = \sum_{i=0}^{n-1} \frac{m}{(1+p)^i} = m\sum_{i=0}^{n-1} x^i\,, \tag{13.57}$$

where $x = \frac{1}{(1+p)}$.

THEOREM 13.30 **Geometric Sum**

$$\forall n \in \mathbb{N}, x \neq 1. \sum_{i=0}^{n} x^i = \frac{1 - x^{n+1}}{1 - x}. \tag{13.58}$$

Proof. Define

$$S = \sum_{i=0}^{n} x^i. \tag{13.59}$$

We compute S by using the perturbation method.

The perturbation would be

$$xS = \sum_{i=0}^{n} x^{i+1} = \sum_{i=1}^{n+1} x^i. \tag{13.60}$$

If we were to subtract xS from S, there would be massive cancellation:

$$S - xS = \sum_{i=0}^{n} x^i - \sum_{i=1}^{n+1} x^i = x^0 + \sum_{i=1}^{n} x^i - \sum_{i=0}^{n} x^i - x^{n+1} = 1 - x^{n+1}. \tag{13.61}$$

$$\therefore S = \frac{1 - x^{n+1}}{1 - x}. \tag{13.62}$$

□

Then we can derive V:

$$V = mS = m\left(\frac{1 + p - \frac{1}{(1+p)^n}}{p}\right). \tag{13.63}$$

13.5.2 Infinite Geometric Series

THEOREM 13.31 **Infinite Geometric Series**

$$\forall |x| < 1. \sum_{i=0}^{\infty} x^i = \frac{1}{1 - x}. \tag{13.64}$$

Proof.

$$\sum_{i=0}^{\infty} x^i = \lim_{n \to \infty} \sum_{i=0}^{n} x^i = \lim_{n \to \infty} \frac{1 - x^{n+1}}{1 - x} = \frac{1}{1 - x}. \tag{13.65}$$

□

In our annuity problem $x = \frac{1}{1+p} < 1$, so we get

$$V = m\sum_{i=0}^{\infty} x^i = m\frac{1}{1-x} = m\frac{1+p}{p}. \tag{13.66}$$

COROLLARY 13.32

$$0.9999\cdots = 1. \tag{13.67}$$

Proof.

$$0.9999\cdots = 0.9\sum_{i=0}^{\infty}(\frac{1}{10})^i = 0.9\frac{1}{1-\frac{1}{10}} = 0.9\frac{10}{9} = 1. \tag{13.68}$$

□

13.5.3 Variations of Geometric Sums

By integrating or differentiating the formulas above, additional formulas arise.

THEOREM 13.33

$$\forall n \in \mathbb{N}, x \neq 1.\ \sum_{i=1}^{n} ix^i = \frac{x-(n+1)x^{n+1}+nx^{n+2}}{(1-x)^2}. \tag{13.69}$$

Proof. By **differentiating or integrating method**. Because

$$\sum_{i=0}^{n} x^i = \frac{1-x^{n+1}}{1-x}, \tag{13.70}$$

we can take derivatives to both sides, i.e.,

$$\frac{\mathrm{d}}{\mathrm{d}x}\sum_{i=0}^{n} x^i = \sum_{i=0}^{n} ix^{i-1} = \sum_{i=1}^{n} ix^{i-1}, \tag{13.71}$$

$$\frac{\mathrm{d}}{\mathrm{d}x}\frac{1-x^{n+1}}{1-x} = \frac{x-(n+1)x^n+nx^{n+1}}{(1-x)^2} \tag{13.72}$$

$$\therefore \sum_{i=1}^{n} ix^{i-1} = \frac{x-(n+1)x^n+nx^{n+1}}{(1-x)^2}. \tag{13.73}$$

We multiply both sides by x

$$\sum_{i=1}^{n} ix^i = \frac{x-(n+1)x^{n+1}+nx^{n+2}}{(1-x)^2}. \tag{13.74}$$

□

COROLLARY 13.34

$$\forall |x| < 1.\ \sum_{i=1}^{\infty} i x^i = \frac{x}{(1-x)^2}\,. \tag{13.75}$$

Proof.

$$\sum_{i=1}^{\infty} i x^i = \lim_{n\to\infty} \sum_{i=1}^{n} i x^i \tag{13.76}$$

$$= \lim_{n\to\infty} \frac{x - (n+1)x^{n+1} + n x^{n+2}}{(1-x)^2} \tag{13.77}$$

$$= \frac{x - \lim_{n\to\infty}(n+1)x^{n+1} + \lim_{n\to\infty} n x^{n+2}}{(1-x)^2} \tag{13.78}$$

$$= \frac{x}{(1-x)^2}\,. \tag{13.79}$$

□

Suppose that there is an annuity that pays im dollars at the end of each year i forever. The value V of this annuity is

$$V = \sum_{i=1}^{\infty} \frac{im}{(1+p)^i} = m\frac{\frac{1}{1+p}}{(1-\frac{1}{1+p})^2} = m\frac{1+p}{p}\,. \tag{13.80}$$

Even though the payments increase every year, the increase is only additive with time; by contrast, dollars paid out in the future decrease in value exponentially with time. The geometric decrease swamps out the additive increase. Payments in the distant future are almost worthless, so the value of the annuity is finite.

13.6 Harmonic Sum and Series

13.6.1 The Book Stacking Problem

Suppose you have a bunch of books (each has length 1) and you want to stack them up, one on top of another in some off-center way, so the top book sticks out past books below it without falling over. If you moved the stack to the edge of a table, how far past the edge of the table do you think you could get the top book to go?

Let's define the *maximum overhang* B_n of a stable stack of n books to be the horizontal distance from the center of mass of the stack to the furthest edge of the top book. If we place the center of mass of the stable stack at the edge of the table, the maximum overhang is how far we can get the top book in the stack to stick out past the edge. We will approach this problem recursively.

Base case: $n = 1$.

$$B_1 = \frac{1}{2}. \tag{13.81}$$

That is to say, if we get one book to stick out, it will not tip as long as its center of mass is over the table.

Inductive step: Now suppose we have a stable stack of $n+1$ books with maximum overhang. If the overhang of the n books on top of the bottom book was not maximum, we could get a book to stick out further by replacing the top stack with a stack of n books with larger overhang. So the maximum overhang B_{n+1} of a stack of $n+1$ books is obtained by placing a maximum overhang stable stack of n books on top of the bottom book. And we get the biggest overhang for the stack of $n+1$ books by placing the center of mass of the n books right over the edge of the bottom book as in Fig. 13.1.

Suppose the difference between the center of mass of the top n books and the center of mass of the whole $(n+1)$ books is r. Then in order to make a balance between the top n books and the bottom book,

$$1 \cdot (\frac{1}{2} - r) = n \cdot r. \tag{13.82}$$

$$\therefore\ r = \frac{1}{2(n+1)}. \tag{13.83}$$

In other words,

$$B_{n+1} - B_n = \frac{1}{2(n+1)}. \tag{13.84}$$

$$\therefore\ B_n = \sum_{i=2}^{n}(B_i - B_{i-1}) + B_1 = \sum_{i=2}^{n}\frac{1}{2i} + \frac{1}{2} = \frac{1}{2}\sum_{i=1}^{n}\frac{1}{i} = \frac{1}{2}H_n. \tag{13.85}$$

13.6.2 Harmonic Number

DEFINITION 13.35 **n-th Harmonic Number H_n**

$$H_n = \sum_{i=1}^{n}\frac{1}{i}. \tag{13.86}$$

There is no known closed-form expression for the harmonic numbers. We can use integration bounds on H_n.

THEOREM 13.36

$$\log n + \frac{1}{n} \le H_n \le \log n + 1. \tag{13.87}$$

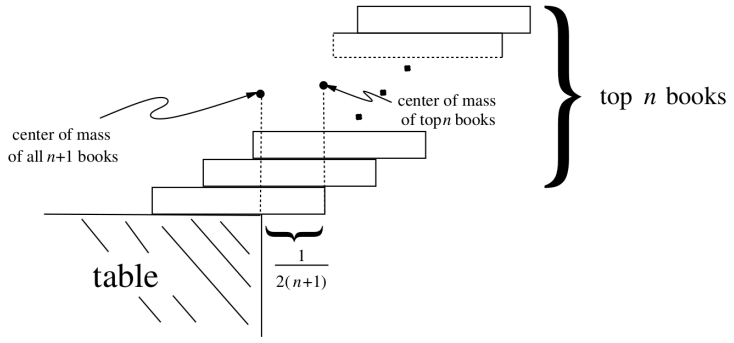

Figure 13.1
Overhanging the edge of the table.

Proof.

$$\int_1^n \frac{1}{x}\,\mathrm{d}x = \log x\Big|_1^n = \log n\,. \tag{13.88}$$

□

COROLLARY 13.37

$$H_n \sim \log n\,. \tag{13.89}$$

Proof.

$$\lim_{n\to\infty} \frac{H_n}{\log n} = 1\,. \tag{13.90}$$

□

THEOREM 13.38

$$H_n = \log n + \gamma + \frac{1}{2n} + \frac{1}{12n^2} + \frac{\varepsilon(n)}{120n^4} = \log n + O(1)\,, \tag{13.91}$$

where $\gamma = 0\ 577215664$ is called *Euler's constant*, and $\varepsilon(n)$ is between 0 and 1 for all n.

For $n > 1$, there are two and only two ways of building a stable stack of n books which both extend the maximum overhang distance: one way where the top book is furthest out, and another way where the second from the top book is furthest out.

By building structures in which more than one book can rests on top of another book, like an inverted pyramid, it is possible to get a stack of n books to extend proportional to $\sqrt[3]{n}$ — much more than $\log n$.

13.7 Double Summations

When there's no obvious closed form for the inner sum, a special trick that is often useful is to try exchanging the order of summation.

THEOREM 13.39

$$\forall n \in \mathbb{N}.\ \sum_{k=1}^{n} H_k = (n+1)H_n - n\,. \tag{13.92}$$

Proof.

$$\sum_{k=1}^{n} H_k = \sum_{k=1}^{n}\sum_{i=1}^{k} \frac{1}{i} \tag{13.93}$$

$$= \sum_{i=1}^{n}\sum_{k=i}^{n} \frac{1}{i} \tag{13.94}$$

$$= \sum_{i=1}^{n} \frac{1}{i}\sum_{k=i}^{n} 1 \tag{13.95}$$

$$= \sum_{i=1}^{n} \frac{1}{i}(n-i+1) \tag{13.96}$$

$$= \sum_{i=1}^{n} \frac{n+1}{i} - \sum_{i=1}^{n} 1 \tag{13.97}$$

$$= (n+1)\sum_{i=1}^{n} \frac{1}{i} - n \tag{13.98}$$

$$= (n+1)H_n - n\,. \tag{13.99}$$

□

13.8 Bounding Summations

It is not always possible to find a closed-form expression for a sum. In such cases, we need to resort to approximations for the sum if we want to have a closed form.

13.8.1 Mathematical Induction

Besides guessing the exact value of a summation in order to use induction, we can use induction to prove a bound on a summation.

THEOREM 13.40

$$\sum_{i=0}^{n} 3^i = O(3^n)\,. \tag{13.100}$$

Proof. By induction. Let inductive hypothesis $P(n)$ be

$$\sum_{i=0}^{n} 3^i \leq c3^n\,, \tag{13.101}$$

for some constant $c \in \mathbb{R}$.

Base case: $n = 0$.

$$\sum_{i=0}^{0} 3^i = 3^0 = 1 \leq c\,, \tag{13.102}$$

if $c \geq 1$.

Inductive step: suppose $P(n)$ is true, then

$$\sum_{i=0}^{n+1} 3^i = \sum_{i=0}^{n} 3^i + 3^{n+1} \leq c3^n + 3^{n+1} = \left(\frac{1}{3} + \frac{1}{c}\right) c3^{n+1} \leq c3^{n+1}\,. \tag{13.103}$$

iff $\frac{1}{3} + \frac{1}{c} \leq 1$, or equivalently, $c \geq \frac{3}{2}$. □

13.8.2 Bounding the Terms

We can obtain a upper bound of a sum by bounding each term using the largest term.

THEOREM 13.41

$$\sum_{i=1}^{n} a_i \leq n \cdot \max_{1 \leq i \leq n} a_i\,. \tag{13.104}$$

Proof.

$$\sum_{i=1}^{n} a_i \leq \sum_{i=1}^{n} \max_{1 \leq j \leq n} a_j = n \cdot \max_{1 \leq j \leq n} a_j\,. \tag{13.105}$$

□

This technique is a weak method when the sum can in fact be bounded by an infinite decreasing geometric series.

THEOREM 13.42 For a strictly decreasing sequence $(a_1, a_2, \ldots, a_n)$, suppose that there exists a constant $0 < r < 1$, such that

$$\forall i \in [1, n-1] . \frac{a_{i+1}}{a_i} \leq r . \tag{13.106}$$

Therefore

$$\sum_{i=1}^{n} a_i \leq \frac{a_1}{1-r} . \tag{13.107}$$

Proof.

$$\sum_{i=1}^{n} a_i \leq \sum_{i=1}^{n} a_1 r^{i-1} = a_1 \sum_{i=0}^{n-1} r^i \leq a_1 \sum_{i=0}^{\infty} r^i = \frac{a_1}{1-r} . \tag{13.108}$$

□

13.8.3 Splitting Summations

One way to obtain bounds on a difficult summation is to express the series as the sum or more series by partitioning the range of the index and then to bound each of the resulting sum.

THEOREM 13.43

$$\sum_{i=1}^{n} i = \quad (n^2) . \tag{13.109}$$

Proof.

$$\sum_{i=1}^{n} i = \sum_{i=1}^{\frac{n}{2}} i + \sum_{i=\frac{n}{2}+1}^{n} i \geq \sum_{i=1}^{\frac{n}{2}} 0 + \sum_{i=\frac{n}{2}+1}^{n} \frac{n}{2} = \left(\frac{n}{2}\right)^2 = \quad (n^2) . \tag{13.110}$$

□

THEOREM 13.44 For a sequence $(a_1, a_2, \ldots, a_n)$, if each a_i is independent of n, then

$$\forall i_0 \in \mathbb{N}^+ . \sum_{i=1}^{n} a_i = \sum_{i=i_0}^{n} a_i + \Theta(1) . \tag{13.111}$$

In other words, we can ignore a constant number of the initial terms. This property also applies to infinite summations.

Proof.

$$\sum_{i=1}^{n} a_i = \sum_{i=1}^{i_0-1} a_i + \sum_{i=i_0}^{n} a_i = \Theta(1) + \sum_{i=i_0}^{n} a_i\,, \tag{13.112}$$

since the initial terms of the summation are all constant and there are a constant number of them. □

THEOREM 13.45

$$\sum_{i=0}^{\infty} \frac{i^2}{2^i} = O(1)\,. \tag{13.113}$$

Proof. We observe that the ratio of consecutive terms is

$$\frac{\frac{(i+1)^2}{2^{i+1}}}{\frac{i^2}{2^i}} = \frac{(i+1)^2}{2i^2} \leq \frac{8}{9} \tag{13.114}$$

if $i \geq 3$. Thus, the summation can be split into

$$\sum_{i=0}^{\infty} \frac{i^2}{2^i} = \sum_{i=3}^{\infty} \frac{i^2}{2^i} + \Theta(1) \leq \frac{9}{8} \sum_{i=0}^{\infty} \left(\frac{8}{9}\right)^i + \Theta(1) = \frac{\frac{9}{8}}{1-\frac{8}{9}} + \Theta(1) = O(1) \tag{13.115}$$

□

THEOREM 13.46

$$H_n = \sum_{i=1}^{n} \frac{1}{i} = O(\log n)\,. \tag{13.116}$$

Proof. We do so by splitting the range 1 to n into $\lfloor \lg n \rfloor + 1$ pieces, the i-th piece consists of the terms in the range $\left[\frac{1}{2^i}, \frac{1}{2^{i+1}}\right)$. The last piece might contain terms not in the original harmonic sum, and we upper bound the contribution of each piece.

$$H_n \leq \sum_{i=0}^{\lfloor \lg n \rfloor} \sum_{j=0}^{2^i-1} \frac{1}{2^i + j} \leq \sum_{i=0}^{\lfloor \lg n \rfloor} \sum_{j=0}^{2^i-1} \frac{1}{2^i} = \sum_{i=0}^{\lfloor \lg n \rfloor} 1 \leq \lg n + 1\,. \tag{13.117}$$

□

13.8.4 Increasing and Decreasing Functions

DEFINITION 13.47 **Strictly Increasing Functions** A function $f\ \mathbb{R}^+ \mapsto \mathbb{R}^+$ is strictly increasing when

$$x_1 < x_2 \Rightarrow f(x_1) < f(x_2)\,. \tag{13.118}$$

DEFINITION 13.48 **Weakly Increasing Functions** A function $f\ \mathbb{R}^+ \mapsto \mathbb{R}^+$ is weakly increasing when

$$x_1 < x_2 \Rightarrow f(x_1) \leq f(x_2)\,. \tag{13.119}$$

DEFINITION 13.49 **Strictly Decreasing Functions** A function $f\ \mathbb{R}^+ \mapsto \mathbb{R}^+$ is strictly decreasing when

$$x_1 < x_2 \Rightarrow f(x_1) > f(x_2)\,. \tag{13.120}$$

DEFINITION 13.50 **Strictly Decreasing Functions** A function $f\ \mathbb{R}^+ \mapsto \mathbb{R}^+$ is strictly decreasing when

$$x_1 < x_2 \Rightarrow f(x_1) \geq f(x_2)\,. \tag{13.121}$$

13.8.5 Integration Bounds

THEOREM 13.51 Let $f\ \mathbb{R}^+ \mapsto \mathbb{R}^+$ be a weakly increasing function, then

$$\int_1^n f(x)\,\mathrm{d}x + f(1) \leq \sum_{i=1}^{n} f(i) \leq \int_1^n f(x)\,\mathrm{d}x + f(n)\,. \tag{13.122}$$

Proof. Suppose $f\ \mathbb{R}^+ \mapsto \mathbb{R}^+$ is a weakly increasing function. $\sum_{i=1}^{n} f(i)$ is the sum of the areas of n unit-width rectangles of heights $f(1), f(2), \ldots, f(n)$, showed in Fig. 13.2, 13.3.

The value of $\int_1^n f(x)\,\mathrm{d}x$ is the shaded area under the curve of $f(x)$ from 1 to n, showed in Fig. 13.2. It is showed that $\sum_{i=1}^{n} f(i)$ is at least $\int_1^n f(x)\,\mathrm{d}x$ plus the area of the leftmost rectangle. Hence

$$\int_1^n f(x)\,\mathrm{d}x + f(1) \leq \sum_{i=1}^{n} f(i)\,. \tag{13.123}$$

Fig. 13.3 shows the curve of $f(x)$ from 1 to n shifted left by 1. It is showed that $\sum_{i=1}^{n} f(i)$ is at most $\int_0^{n-1} f(x+1)\,\mathrm{d}x$ plus the area of the rightmost rectangle. Hence,

$$\sum_{i=1}^{n} f(i) \leq \int_0^{n-1} f(x+1)\,\mathrm{d}x + f(n) = \int_1^n f(x)\,\mathrm{d}x + f(n)\,. \tag{13.124}$$

□

THEOREM 13.52 Let $f\ \mathbb{R}^+ \mapsto \mathbb{R}^+$ be a weakly decreasing function, then

$$\int_1^n f(x)\,\mathrm{d}x + f(1) \geq \sum_{i=1}^{n} f(i) \geq \int_1^n f(x)\,\mathrm{d}x + f(n)\,. \tag{13.125}$$

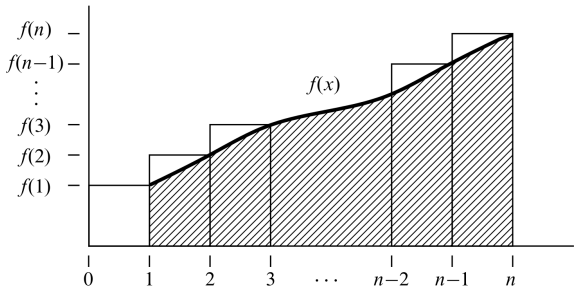

Figure 13.2
The shaded area under the curve of $f(x)$ from 1 to n (shown in bold) is $\int_1^n f(x)\,\mathrm{d}x$.

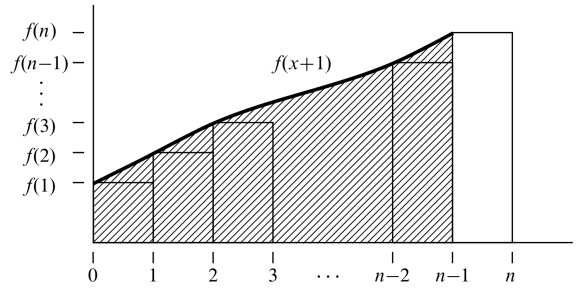

Figure 13.3
This curve is the same as the curve in Fig. 13.2 shifted left by 1.

Proof. By a similar argument using Fig. 13.4.

□

13.9 Dealing with Products

13.9.1 Product Formulas

DEFINITION 13.53 **Finite Product** Given a sequence $(a_1, a_2, \ldots, a_n)$ of numbers where $n \in \mathbb{N}$,

$$\prod_{i=1}^{n} a_i := a_1 \cdot a_2 \cdot \cdots \cdot a_n \,. \tag{13.126}$$

If $n = 0$, the value of summation is defined to be 1.

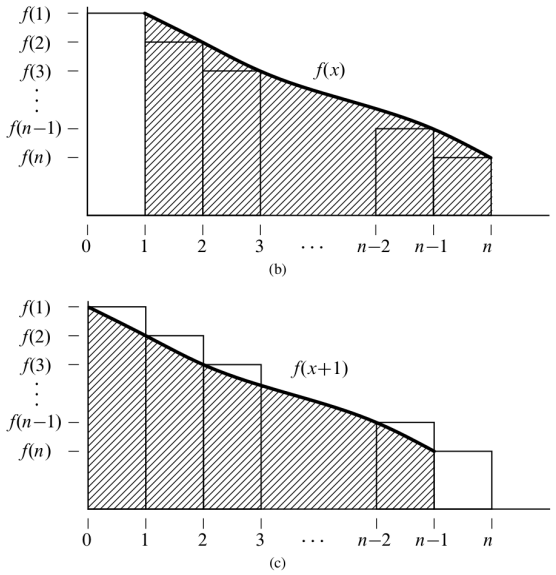

Figure 13.4
The shaded area under the curve of $f(x)$ from 1 to n (shown in bold) is $\int_1^n f(x)\,\mathrm{d}x$.

DEFINITION 13.54 **Infinite Product** Given a sequence $(a_1, a_2, \ldots)$ of numbers,

$$\prod_{i=1}^{\infty} a_i := \lim_{n\to\infty} \prod_{i=1}^{n} a_i = a_1 \cdot a_2 \cdot \cdots . \tag{13.127}$$

LEMMA 13.55 [Product to Summation Conversion]

$$\log \prod_{i=1}^{n} a_i = \sum_{i=1}^{n} \log a_i \,. \tag{13.128}$$

13.9.2 Factorials

THEOREM 13.56

$$\frac{n^n}{\exp(n-1)} \le n! \le \frac{n^{n+1}}{\exp(n-1)} \,. \tag{13.129}$$

Proof. Convert any product into a sum by taking a logarithm.

$$\log n! = \log \prod_{i=1}^{n} i = \sum_{i=1}^{n} \log i \tag{13.130}$$

Apply the integration bounds.

$$\int_{x=1}^{n} \log x \, \mathrm{d}x = (x \log x - x)\Big|_1^n = n \log n - n + 1 \,. \tag{13.131}$$

$$\therefore \; n \log n - n + 1 \leq \sum_{i=1}^{n} \log i \leq n \log n - n + 1 + \log n = (n+1) \log n - n + 1 \tag{13.132}$$

Exponentiate on both sides

$$\frac{n^n}{\exp(n-1)} \leq n! \leq \frac{n^{n+1}}{\exp(n-1)} \,. \tag{13.133}$$

□

13.9.3 Stirling's Formula

THEOREM 13.57 **Stirling's Formula**

$$\forall n \geq 1. \; n! = \sqrt{2\pi n} \left(\frac{n}{e}\right)^n \exp \varepsilon(n) \,, \tag{13.134}$$

where $\frac{1}{12n+1} \leq \varepsilon(n) \leq \frac{1}{12n}$.

COROLLARY 13.58

$$\forall n \geq 1. \; n! \sim \sqrt{2\pi n} \left(\frac{n}{e}\right)^n \,. \tag{13.135}$$

Proof.

$$\lim_{n \to \infty} \frac{n!}{\sqrt{2\pi n} \left(\frac{n}{e}\right)^n} = \lim_{n \to \infty} \exp \varepsilon(n) = \exp 0 = 1 \,. \tag{13.136}$$

□

14 Recurrences

DEFINITION 14.1 **Recurrence** A recurrence describes a sequence of numbers. Early terms are specified explicitly, and later terms are expressed as a function of their predecessors.

Generating smaller subproblems is far more important to algorithmic speed than reducing the additional steps per recursive call. More generally, linear recurrences typically have exponential solutions, while divide-and-conquer recurrences usually have solutions bounded by a polynomial.

The number of subproblems will affect the solution for both linear and divide-and-conquer recurrences. Boundary conditions are almost irrelevant for divide-and-conquer recurrences, and the solutions of linear recurrence are usually dominated by an exponential whose base is determined by the number and size of subproblems. Boundary conditions matter greatly only when they give the dominant term a zero coefficient, which changes the asymptotic solution.

14.1 The Towers of Hanoi

14.1.1 A Recursive Solution

We want to know how many moves do we need in order to move all the disks to one of the other two posts on top. Let's define $f(n)$ to be the minimum number of moves for n disks.

Base case: $f(1) = 1$.

Inductive step step: There are three steps to move n disks.

1. Move the top $(n-1)$ disks from the first post to the second using the solution for $(n-1)$ disks. This can be done in $f(n-1)$ steps.
2. Move the largest disk from the first post to the third post. This takes just 1 step.
3. Move the $(n-1)$ disks from the second post to the third post, again using the solution for $(n-1)$ disks. This can also be done in $f(n-1)$ steps.

Therefore

$$f(n) \leq 2f(n-1) + 1\,. \tag{14.1}$$

$2f(n-1)+1$ is actually a lower bound of $f(n)$. At some step, we must must move the largest disk from the first post to a different post. For this to happen, the $(n-1)$ smaller disks must all be stacked out of the way on the only remaining post. Arranging the $(n-1)$ smaller disks this way requires at least $f(n-1)$ moves. After the largest disk is moved, at least another $f(n-1)$ moves are required to pile the $(n-1)$ smaller disks on top.

$$\therefore f(n) = \begin{cases} 1 & \text{if } n = 1 \\ 2f(n-1)+1 & \text{if } n \geq 2 \end{cases}. \tag{14.2}$$

14.1.2 Guess and Verify/Substitution Methods

There are several methods for solving recurrence equations. The simplest is to guess the solution by computing several terms at the beginning of the sequence $f(1), f(2), f(3)$, etc., and then verify that the guess is correct with an induction proof.

THEOREM 14.2 $f(n) = 2^n - 1$ satisfies the Hanoi recurrence.

Proof. By induction on n. The inductive hypothesis

$$P(n)\ f(n) = 2^n - 1\,. \tag{14.3}$$

Base case: $n = 1$.

$$f(1) = 1 = 2^1 - 1\,. \tag{14.4}$$

Inductive step: Assume $f(n) = 2^n - 1$, then

$$f(n+1) = 2f(n) + 1 = 2(2^n - 1) + 1 = 2^{n+1} - 1\,. \tag{14.5}$$

□

14.1.3 Plug and Chug/Expansion/Iteration

Plug-and-chug is another way to solve recurrences when it is difficult to guess from a strange recurrence form. As in guess-and-verify, the key step is identifying a pattern. But instead of looking at a sequence of numbers, you have to spot a pattern in a sequence of expressions, which is sometimes easier. It has three steps.

1. Expand the recurrence equation by alternately "plugging" (applying the recurrence) and "chugging" (simplifying the result) until a pattern appears. Be careful: too much simplification can make a pattern harder to spot.
2. Verify the general formula with one more round of plug-and-chug.
3. Write $f(n)$ using early terms with known values.

For example, for the Hanoi recurrence:

$$f(n) = 1 + 2f(n-1) \tag{14.6}$$
$$= 1 + 2(1 + 2f(n-2)) \quad \text{(plug)} \tag{14.7}$$
$$= 1 + 2 + 4f(n-2) \quad \text{(chug)} \tag{14.8}$$
$$= 1 + 2 + 4(1 + 2f(n-3)) \quad \text{(plug)} \tag{14.9}$$

$$= 1 + 2 + 4 + 8f(n-3) \quad \text{(chug)} \tag{14.10}$$
$$= 1 + 2 + 4 + \cdot + 2^i f(n-i) \quad \text{(A pattern appears)} \tag{14.11}$$
$$= 2^i - 1 + 2^i f(n-i) \quad \text{(Simplify)}. \tag{14.12}$$

Then we need to verify the pattern:

$$f(n) = 2^i - 1 + 2^i f(n-i) \tag{14.13}$$
$$= 2^i - 1 + 2^i(1 + 2f(n-(i+1))) \quad \text{(plug)} \tag{14.14}$$
$$= 2^{i+1} - 1 + 2^{i+1} f(n-(i+1)) \quad \text{(chug)}. \tag{14.15}$$

Finally, we choose $i = n-1$

$$f(n) = 2^{n-1} - 1 + 2^{n-1} f(1) = 2^n - 1. \tag{14.16}$$

14.2 Merge Sort

14.2.1 Merge Sort

Suppose we want to sort an array of n numbers. There are two cases:

- If the input is a single number, then the algorithm does nothing, because the list is already sorted.
- Otherwise, the list contains two or more numbers. The first half and the second half of the list are each sorted recursively. Then the two halves are merged to form a sorted list with all n numbers.

Merging is done by repeatedly emitting the smaller of the two leading terms. When one list is empty, the whole other list is emitted.

14.2.2 Recurrence Solution

We want to know what the maximum number of comparisons used in sorting n items is. This is taken as an estimate of the running time. Let $f(n)$ be the maximum number of comparisons used while Merge Sorting a list of n numbers. For now, assume that n is a power of 2.

Base case: If there is only one number in the list, then no comparisons are required, so $f(1) = 0$.

Inductive step: $f(n)$ includes comparisons used in sorting the first half (at most $f(\frac{n}{2})$), in sorting the second half (also at most $f(\frac{n}{2})$), and in merging the two halves. The number of comparisons in the merging step is at most $n-1$. This is because at least one number is emitted after each comparison and one more number is emitted at the end when one list becomes empty.

Therefore $f(n)$ can be computed by this recurrence

$$f(n) = \begin{cases} 0 & \text{if } n = 1 \\ 2f(\frac{n}{2}) + n - 1 & \text{if } n \geq 2 \text{ and } n \text{ is a power of } 2 \end{cases}. \tag{14.17}$$

14.2.3 Plug and Chug

Guessing the solution to this recurrence is hard because there is no obvious pattern from first few values. So let's try the plug-and-chug method instead.

$$\begin{aligned} f(n) &= n - 1 + 2f(\frac{n}{2}) && (14.18)\\ &= n - 1 + 2\left(\frac{n}{2} - 1 + 2f(\frac{n}{4})\right) && (14.19)\\ &= n - 1 + n - 2 + 4f(\frac{n}{4}) && (14.20)\\ &= n - 1 + n - 2 + 4\left(\frac{n}{4} - 1 + 2f(\frac{n}{8})\right) && (14.21)\\ &= n - 1 + n - 2 + n - 4 + 8f(\frac{n}{8}) && (14.22)\\ &= n - 2^0 + n - 2^1 + n - 2^2 + \cdots + n - 2^{i-1} + 2^i f(\frac{n}{2^i}) && (14.23)\\ &= in - 2^i + 1 + 2^i f(\frac{n}{2^i}) && (14.24) \end{aligned}$$

Next, we verify the pattern with one additional round of plug-and-chug.

$$\begin{aligned} f(n) &= in - 2^i + 1 + 2^i f(\frac{n}{2^i}) && (14.25)\\ &= in - 2^i + 1 + 2^i\left(\frac{n}{2^i} - 1 + 2f(\frac{n}{2^{i+1}})\right) && (14.26)\\ &= (i+1)n - 2^{i+1} + 1 + 2^{i+1} f(\frac{n}{2^{i+1}}). && (14.27) \end{aligned}$$

Finally, we express $f(n)$ using early terms whose values are known. Specifically, if we let $i = \lg n$,

$$\begin{aligned} f(n) &= in - 2^i + 1 + 2^i f(\frac{n}{2^i}) && (14.28)\\ &= n\lg n - 2^{\lg n} + 1 + 2^{\lg n} f(\frac{n}{2^{\lg n}}) && (14.29)\\ &= n\lg n - n + 1 + nf(1) && (14.30)\\ &= n\lg n - n + 1. && (14.31) \end{aligned}$$

14.3 Linear Recurrences

14.3.1 Climbing Stairs

How many different ways are there to climb n stairs, if you can either step up one stair or hop up two? Let $f(n)$ denotes the number of ways to climb n stairs.

Base cases: there is 1 way to climb 0 stairs (do nothing) and 1 way to climb 1 stair (step up), i.e.,

$$f(0) = 1 \tag{14.32}$$

$$f(1) = 1\,. \tag{14.33}$$

Inductive step: an ascent of n stairs consists of either a step followed by an ascent of the remaining $(n-1)$ stairs or a hop followed by an ascent of $(n-2)$ stairs. So the total number of ways to climb n stairs is equal to the number of ways to climb $(n-1)$ plus the number of ways to climb $(n-2)$.

$$f(n) = f(n-1) + f(n-2)\,. \tag{14.34}$$

Therefore $f(n)$ can be computed by this recurrence

$$f(n) = \begin{cases} 1 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ f(n-1) + f(n-2) & \text{if } n \geq 2 \end{cases}. \tag{14.35}$$

$f(n)$ is actually the n-th Fibonacci number.

14.3.2 Solving Homogeneous Linear Recurrences

DEFINITION 14.3 **Homogeneous Linear Recurrence** A homogeneous linear recurrence has the form

$$f(n) = \sum_{i=1}^{d} a_i f(n-i)\,. \tag{14.36}$$

The **order** of the recurrence is d. The value of the function $f(n)$ is also specified at a few points, which are called **boundary conditions**.

In general, linear recurrences tend to have exponential solutions. Substituting the guess $f(n) = x^n$ gives

$$x^n = \sum_{i=1}^{d} a_i x^{n-i}\,. \tag{14.37}$$

Dividing by x^{n-d} and arranging the items gives

$$x^d - \sum_{i=1}^{d} a_i x^{d-i} = 0\,. \tag{14.38}$$

This is called the **characteristic equation** of the recurrence. The solutions $x_1, x_2, \ldots, x_d$ to a linear recurrence are defined by the roots of the characteristic equation, Neglecting boundary conditions for the moment:

- If x is a nonrepeated root of the characteristic equation, then x^n is a solution to the recurrence.
- If x is a repeated root with multiplicity k then $x^n, nx^n, n^2x^n, \ldots, n^{k-1}x^n$ are all solutions to the recurrence.

THEOREM 14.4 If $f_1(n), f_2(n), \ldots, f_d(n)$ are both solutions to a homogeneous linear recurrence, then $\sum_{i=1}^{d} t_i f_i(n)$ is also a solution for all $t_i \in \mathbb{R}$.

Proof.

$$\sum_{i=1}^{d} t_i f_i(n) = \sum_{i=1}^{d} t_i \sum_{j=1}^{d} a_j f_i(n-i) \tag{14.39}$$

$$= \sum_{i=1}^{d} \sum_{j=1}^{d} t_i\, a_j f_i(n-i) \tag{14.40}$$

$$= \sum_{j=1}^{d} \sum_{i=1}^{d} t_i\, a_j f_i(n-i) \tag{14.41}$$

$$= \sum_{j=1}^{d} a_j \sum_{i=1}^{d} t_i f_i(n-i)\,. \tag{14.42}$$

□

The above theorem implies that every linear combination of these solutions is also a solution. So the solution will be

$$f(n) = \sum_{i=1}^{d} t_i x_i^n\,. \tag{14.43}$$

All that remains is to select a solution consistent with the boundary conditions by choosing the constants appropriately. We can determine the constants by solving a system of linear equations

$$f(u) = \sum_{i=1}^{d} t_i\, x_i^u, \forall u = 1, 2, \ldots, d\,. \tag{14.44}$$

14.3.3 Solving the Fibonacci Recurrence

Substituting $f(n) = x^n$ gives the characteristic equation

$$x^2 - x - 1 = 0\,. \tag{14.45}$$

Solving this equation gives two roots

$$x_1 = \frac{1+\sqrt{5}}{2} \tag{14.46}$$

$$x_2 = \frac{1-\sqrt{5}}{2}\,. \tag{14.47}$$

So the solution has form

$$f(n) = t_1\left(\frac{1+\sqrt{5}}{2}\right)^n + t_2\left(\frac{1-\sqrt{5}}{2}\right)^n\,. \tag{14.48}$$

We can find the solution that satisfies the boundary conditions

$$f(0) = t_1\left(\frac{1+\sqrt{5}}{2}\right)^0 + t_2\left(\frac{1-\sqrt{5}}{2}\right)^0 = 1 \tag{14.49}$$

$$f(1) = t_1\left(\frac{1+\sqrt{5}}{2}\right)^1 + t_2\left(\frac{1-\sqrt{5}}{2}\right)^1 = 1\,. \tag{14.50}$$

The final solution is

$$f(n) = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^{n+1} - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^{n+1}\,. \tag{14.51}$$

This closed form for Fibonacci numbers is known as **Binet's formula**.

DEFINITION 14.5 **Golden Ratio ϕ**

$$\phi = \frac{1+\sqrt{5}}{2} \approx 1.618\,. \tag{14.52}$$

COROLLARY 14.6 The n-th Fibonacci number $f(n)$ is

$$f(n) = \frac{1}{\sqrt{5}}\phi^{n+1} + o(1)\,. \tag{14.53}$$

Proof.

$$\lim_{n\to\infty}\left|\left(\frac{1-\sqrt{5}}{2}\right)^n\right| \approx |(-0.618)^n| = 0\,. \tag{14.54}$$

□

COROLLARY 14.7 The ratio of consecutive Fibonacci numbers rapidly approaches the golden ratio:

$$\forall i. \frac{f(i+1)}{f(i)} \approx \phi . \tag{14.55}$$

Proof.

$$\frac{f(i+1)}{f(i)} = \frac{\frac{1}{\sqrt{5}}\phi^{i+2} + o(1)}{\frac{1}{\sqrt{5}}\phi^{i+1} + o(1)} \approx \phi \tag{14.56}$$

□

14.3.4 Solving General Linear Recurrences

DEFINITION 14.8 **Inhomogeneous Linear Recurrence** An inhomogeneous linear recurrence has the form

$$f(n) = \sum_{i=1}^{d} a_i f(n-i) + g(n) . \tag{14.57}$$

The **order** of the recurrence is d. The value of the function $f(n)$ is also specified at a few points, which are called **boundary conditions**.

Solving inhomogeneous linear recurrences needs the following steps:

1. Replace $g(n)$ by 0. As before, find the roots of the characteristic equation

$$x^d - \sum_{i=1}^{d} a_i x^{d-i} = 0 . \tag{14.58}$$

2. Write down the homogeneous solution

$$f_n(n) = \sum_{i=1}^{d} t_i x_i^n . \tag{14.59}$$

3. Find a particular solution $f_p(n)$. This is a solution to the full recurrence that need not be consistent with the boundary conditions.
4. Form the general solution, which is the sum of the homogeneous solution and the particular solution

$$f(n) = f_n(n) + f_p(n) = \sum_{i=1}^{d} t_i x_i^n + f_p(n) . \tag{14.60}$$

5. Substitute the boundary conditions into the general solution.

$$f(u) = \sum_{i=1}^{d} t_i x_i^u + f_p(u), \forall u = 1, 2, \ldots, d. \tag{14.61}$$

How to Guess a Particular Solution: Generally, look for a particular solution with the same form as the inhomogeneous term $g(n)$.

- If $g(n)$ is a constant, then guess a particular solution $f_p(n) = c$. If this doesn't work, try polynomials of progressively higher degree: $f_p(n) = bn + c$, then $f_p(n) = an^2 + bn + c$, etc.
- If $g(n)$ is a polynomial, try a polynomial of the same degree, then a polynomial of degree one higher, then two higher, etc. For example, if $g(n) = 6n + 5$, then try $g_p(n) = bn + c$ and then $g(n) = an^2 + bn + c$.
- If $g(n)$ is an exponential, such as 3^n, then first guess that $f_p(n) = c\,3^n$. Failing that, try $f_p(n) = bn\,3^n + c\,3^n$ and then $f_p(n) = an^2 3^n + bn\,3^n + c\,3^n$, etc.

14.3.5 Solving General Linear Recurrences: Example

$$f(n) = \begin{cases} 1 & \text{if } n = 1 \\ 2f(n-1) + n & \text{if } n \geq 2 \end{cases}. \tag{14.62}$$

The characteristic equation is

$$x - 2 = 0. \tag{14.63}$$

So the homogeneous solution is

$$f_n(n) = t\,2^x. \tag{14.64}$$

In Step 3, we must find a particular solution to the full recurrence. Let's guess that there is a solution of the form $f_p(n) = an + b$. Substituting this guess into the recurrence gives

$$an + b = 2(a(n-1) + b) + n \tag{14.65}$$

$$\therefore\ (a+1)n + (b - 2a) = 0. \tag{14.66}$$

$$\therefore \begin{cases} a + 1 & = 0 \\ b - 2a & = 0 \end{cases}. \tag{14.67}$$

$$\therefore \begin{cases} a & = -1 \\ b & = -2 \end{cases}. \tag{14.68}$$

$$\therefore f_p(n) = -n - 2\,. \tag{14.69}$$

The general solution is

$$f(n) = t\,2^n - n - 2\,. \tag{14.70}$$

We use the boundary condition

$$f(1) = t\,2^1 - 1 - 2 = 1\,. \tag{14.71}$$

$$\therefore t = 2\,. \tag{14.72}$$

$$\therefore f(n) = 2^{n+1} - n - 2\,. \tag{14.73}$$

14.4 Divide-and-Conquer Recurrences

14.4.1 Divide-and-Conquer Recurrences

DEFINITION 14.9 **Divide-and-Conquer Recurrence** A divide-and-conquer recurrence has the form

$$f(n) = \sum_{i=1}^{d} a_i f(b_i n) + g(n)\,, \tag{14.74}$$

where $\forall i.\ a_i > 0 \wedge 0 < b_i < 1$, and $g(n)$ is a nonnegative function.

Merge Sort is an example of a divide-and-conquer algorithm: it divides the input, "conquers" the pieces, and combines the results.

The asymptotic solution to a divide-and-conquer recurrence is independent of the boundary conditions, and it is unaffected by floors and ceilings.

14.4.2 The Master Theorem

The Master Theorem handles some of the recurrences that commonly arise in computer science.

THEOREM 14.10 **Master Theorem** Let T be a recurrence of the form

$$f(n) = aT\left(\frac{n}{b}\right) + g(n)\,. \tag{14.75}$$

Case 1: If $g(n) = O(n^{\log_b a - \varepsilon})$ for some constant $\varepsilon > 0$, then

$$f(n) = \Theta(n^{\log_b a})\,. \tag{14.76}$$

Case 2: If $g(n) = O(n^{\log_b a} \cdot \lg^k n)$ for some constant $k \geq 0$, then

$$f(n) = \Theta(n^{\log_b a} \lg^{k+1} n)\,. \tag{14.77}$$

Case 3: If $g(n) = \quad (n^{\log_b a + \varepsilon})$ for some constant $\varepsilon > 0$, and $a \cdot g\left(\frac{n}{b}\right) < c \cdot g(n)$ for some constant $c < 1$ and sufficiently large n, then

$$f(n) = \Theta(g(n))\,. \tag{14.78}$$

Proof. By induction on n. □

14.4.3 The Akra-Bazzi Theorem

All the recurrences we have considered were defined over the integers, and that is the common case. But the Akra-Bazzi theorem applies more generally to functions defined over the real numbers.

THEOREM 14.11 **Akra-Bazzi** Suppose that the function $T\ \mathbb{R} \mapsto \mathbb{R}$ is nonnegative and bounded for $0 \leq x \leq x_0$ and satisfies the recurrence

$$f(x) = \sum_{i=1}^{d} a_i f(b_i x + \varepsilon_i(x)) + g(x) \quad \text{for } x > x_0\,, \tag{14.79}$$

where

- x_0 is large enough so that T is well-defined,
- $\forall i.\ a_i > 0$,
- $\forall i.\ 0 < b_i < 1$,
- $\forall i.\ |\varepsilon_i(x)| = O\left(\frac{x}{\log^2 x}\right)$,
- $g(x)$ is a nonnegative function such that $|g'(x)|$ is bounded by a polynomial.

Then

$$f(x) = \Theta\left(x^p + x^p \int_1^x \frac{g(t)}{t^{p+1}}\,\mathrm{d}t\right)\,, \tag{14.80}$$

where p satisfies

$$\sum_{i=1}^{d} a_i b_i^p = 1 . \tag{14.81}$$

Proof. By a complicated induction. □

$\varepsilon_i(x)$ extends the theorem to address floors, ceilings, and other small adjustments to the sizes of subproblems. These functions $\varepsilon_i(x)$ do not affect, or even appear in, the asymptotic solution to the recurrence.

15 Cardinality Rules

The most direct way to count one thing by counting another is to find a bijection between them, since if there is a bijection between two sets, then the sets have the same size. This important fact is commonly known as the **Bijection Rule**.

15.1 The Sum Rule and the Product Rule

The Bijection Rule suggests a general strategy: get really good at counting just a few things, then use bijections to count everything else. In particular, we will get really good at counting sequences. When we want to determine the size of some other set A, we will find a bijection from A to a set of sequences B. Then we will use our sequence-counting skills to determine $|B|$, which immediately gives us $|A|$.

15.1.1 The Product Rule

15.1.1.1 The Product Rule

THEOREM 15.1 **Product Rule** If $A_1, A_2, \ldots, A_n$ are finite sets, then:

$$|A_1 \times A_2 \times \cdots \times A_n| = |A_1| \cdot |A_2| \cdots |A_n| . \tag{15.1}$$

15.1.1.2 Subsets of an n-element Set

THEOREM 15.2

$$|S| = n \Rightarrow |\operatorname{pow} S| = 2^n . \tag{15.2}$$

Proof. Let $S = \{s_1, s_2, \ldots, s_n\}$. Let

$$A = \{0,1\}^n = \{(a_1, a_2, \ldots, a_n) \mid \forall i.\ a_i \in \{0,1\}\} \tag{15.3}$$

be the set of all n-bit strings. Let $f\ A \mapsto \operatorname{pow} S$ be a bijection that maps a n-bit string $(a_1, a_2, \ldots, a_n)$ to a subset $C \in \operatorname{pow} S$ with

$$\forall i.\ s_i \in C \Leftrightarrow a_i = 1 . \tag{15.4}$$

By the product rule,

$$|\operatorname{pow} A| = |A| = |\{0,1\}^n| = |\{0,1\}|^n = 2^n . \tag{15.5}$$

□

15.1.2 The Sum Rule

15.1.2.1 The Sum Rule

THEOREM 15.3 **Sum Rule** If $A_1, A_2, \ldots, A_n$ are *disjoint* finite sets, then:

$$|A_1 \cup A_2 \cup \cdots \cup A_n| = |A_1| + |A_2| + \cdots + |A_n|. \tag{15.6}$$

15.1.2.2 Counting Passwords

THEOREM 15.4 If a valid password is a sequence of between six and eight symbols. The first symbol must be a letter (which can be lowercase or uppercase), and the remaining symbols must be either letters or digits. There are 1.8×10^{14} different possible passwords.

Proof. Let

$$S_L = \{a, b, \ldots, z, A, B, \ldots, Z\} \tag{15.7}$$

be the set of all letters. Let

$$S_A = \{a, b, \ldots, z, A, B, \ldots, Z, 0, 1, \ldots, 9\} \tag{15.8}$$

be the set of all letters and digits.

The total number of possible passwords is

$$\begin{aligned}
&|(S_L \times S_A^5) \cup (S_L \times S_A^6) \cup (S_L \times S_A^7)| && (15.9)\\
&\quad = |S_L \times S_A^5| + |S_L \times S_A^6| + |S_L \times S_A^7| && (15.10)\\
&\quad = |S_L| \cdot |S_A|^5 + |S_L| \cdot |S_A|^6 + |S_L| \cdot |S_A|^7 && (15.11)\\
&\quad = 52 \times 62^5 + 52 \times 62^6 + 52 \times 62^7 && (15.12)\\
&\quad \approx 1.8 \times 10^{14}. && (15.13)
\end{aligned}$$

□

15.1.3 The Generalized Product Rule

15.1.3.1 The Generalized Product Rule

THEOREM 15.5 **Generalized Product Rule** Let S be a set of length-k sequences. If there are

- n_1 possible first entries,
- n_2 possible second entries for each first entry,
- n_k possible k-th entries for each sequence of first $(k-1)$ entries,

then

$$|S| = n_1 \cdot n_2 \cdots n_k \,. \tag{15.14}$$

15.1.3.2 Defective Dollar Bills

THEOREM 15.6 Let's say that a dollar bill is defective if some digit appears more than once in the 8-digit serial number. Assuming that the digit portions of serial numbers all occur equally often. Then

$$\Pr\{\text{your bill is defective}\} = 98.1856\% \,. \tag{15.15}$$

Proof.

$$\begin{aligned}
\Pr\{\text{your bill is defective}\} &= 1 - \Pr\{\text{your bill is nondefective}\} && (15.16)\\
&= 1 - \frac{|\{\text{serial numbers with all digits different}\}|}{|\{\text{serial numbers}\}|} && (15.17)\\
&= 1 - \frac{10 \times 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3}{10^8} && (15.18)\\
&= 1 - 1.8144\% && (15.19)\\
&= 98.1856\% \,. && (15.20)
\end{aligned}$$

□

15.1.3.3 A Chess Problem

THEOREM 15.7 There are 112,896 ways to place 3 different chess pieces on a chessboard so that no 2 pieces share a row a column.

Proof. Let

$$A = \{(i_1, j_1, i_2, j_2, i_3, j_3) \in [1,8]^6 \mid i_1 \neq i_2 \wedge i_2 \neq i_3 \wedge j_1 \neq j_2 \wedge j_2 \neq j_3\} \,. \tag{15.21}$$

$|A| = 8 \times 8 \times 7 \times 7 \times 6 \times 6 = 112,896$ Let B be the set of all valid chess pieces configurations. There is a bijection $f \;\; A \mapsto B$ maps the sequence $(i_1, j_1, i_2, j_2, i_3, j_3)$ to a configuration

with the first piece in row i_1, column j_1, the second piece in row i_2, column j_2, and the third piece in row i_3, column j_3.

Therefore

$$|B| = |A| = \frac{3,136}{2} = 1,568\,. \tag{15.22}$$

since there are 8 ways to set i_1 and 8 ways to set j_1, 7 remaining ways to set i_2 and 7 remaining ways to set j_2, 6 remaining ways to set i_3 and 6 remaining ways to set j_3. □

15.1.3.4 Permutations

DEFINITION 15.8 **Permutation of a Finite Set** A permutation of a finite set A is a sequence that contains every element of A exactly once.

THEOREM 15.9 There are $n!$ permutations of an n element set.

Proof. The total number of permutations of an n element set is

$$n \cdot (n-1) \cdots 1 = n! \tag{15.23}$$

since there are n choices for the first element, $(n-1)$ remaining choices for the second element, $(n-2)$ remaining choices for the third element for every combination of the first two elements, and so forth. □

DEFINITION 15.10 **k-Permutations** A k-permutation of A is a sequence of k elements of A, with no element appearing more than once in the sequence. (Thus, an ordinary permutation is an n-permutation of an n-set).

THEOREM 15.11 The number of k-permutations of an n-set is

$$\binom{n}{k} k! = \frac{n!}{(n-k)!}\,. \tag{15.24}$$

15.2 The Division Rule

15.2.1 The Division Rule

DEFINITION 15.12 **k-to-1 Function** A k-to-1 function maps exactly k elements of the domain to every element of the codomain.

THEOREM 15.13 **Division Rule** If $f\ A \mapsto B$ is k-to-1, then $|A| = k|B|$.

15.2.2 Two Rooks Problem

THEOREM 15.14 There are 1,568 different ways to place two identical rooks on a chessboard so that they do not share a row or column.

Proof. Let

$$A = \{(i_1, j_1, i_2, j_2) \in [1,8]^4 \mid i_1 \neq i_2 \wedge j_1 \neq j_2\}. \tag{15.25}$$

Let B be the set of all valid rook configurations. There is a natural function $f\ A \mapsto B$ maps the sequence (i_1, j_1, i_2, j_2) to a configuration with one rook in row i_1, column j_1 and the other rook in row i_2, column j_2. Since f maps exactly two sequences to every board configuration, f is a 2-to-1 function.

Therefore

$$|B| = \frac{|A|}{2} = \frac{3,136}{2} = \frac{8 \times 8 \times 7 \times 7}{2} = 1,568, \tag{15.26}$$

since there are 8 ways to set i_1 and 8 ways to set j_1, 7 remaining ways to set i_2 and 7 remaining ways to set j_2. □

15.2.3 Knights of the Round Table

THEOREM 15.15 There are $(n-1)$ different ways for King Arthur to arrange to seat his n different knights at his round table. A seating defines who sits where. Two seatings are considered to be the same arrangement if each knight sits between the same two knights in both seatings.

Proof. Let $|A|$ be the set of all permutations of {1, 2, ..., n}, $|A| = n!$. Let B be the set of all different seatings. There is a natural function $f\ A \mapsto B$ maps the sequence $(\pi(1), \pi(2), \ldots, \pi(n))$ to a seating of knights going clockwise around the table starting at the top seat. f maps exactly n sequences to every seating, f is a n-to-1 function, since all n cyclic shifts of the sequence of knights in the seating map to the same arrangement.

Therefore

$$|B| = \frac{|A|}{n} = \frac{n!}{n} = (n-1)!. \tag{15.27}$$

□

15.3 The Subset Rule

15.3.1 The Subset Rule

DEFINITION 15.16 ***k*-Combinations** A k-combination of an n-set A is a k-subset of A.

THEOREM 15.17 **Subset Rule** The number of k-combinations is

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}. \tag{15.28}$$

$\binom{n}{k}$ is called **Binomial Coefficient**.

Proof. Suppose $S = \{a_1, a_2, \ldots, a_n\}$. Let A be the set of all permutations of the n-element set S. Let B be the set of all possible k-element subset of S. Let f be the mapping that takes the first k elements of the permutation from A to the set.

Note that any other permutation with the same first k elements in any order and the same remaining elements $n-k$ elements in any order will also map to this set. Since there are $k!$ possible permutations of the first k elements and $(n-k)!$ permutations of the remaining elements, the mapping f is $k!(n-k)!$-to-1. Therefore

$$|B| = \binom{n}{k} = \frac{|A|}{k!(n-k)!} = \frac{n!}{k!(n-k)!}. \tag{15.29}$$

□

15.3.2 Bit Sequences

COROLLARY 15.18 The number of n-bit sequences with exactly k ones is $\binom{n}{k}$.

Proof. By constructing a bijection from k-element subsets of an n-element set and n-bit sequence with k ones, and applying the Subset Rule. □

LEMMA 15.19 The number of ways to select m balls when n different types are available is $\binom{m+n-1}{m}$

Proof. Let

$$A = \text{all } (m+n-1)\text{-bit } \{0,1\} \text{ sequences with exactly } m \text{ zeros and } (n-1) \text{ ones} \tag{15.30}$$

Let

$$B = \text{the all ways to select } m \text{ balls when } n \text{ different types are available}. \quad (15.31)$$

Let f be the mapping to m balls consisting of m_i balls of type i from the sequence

$$\underbrace{0\ldots0}_{m_1}\ 1\ \underbrace{0\ldots0}_{m_2}\ 1\ldots1\ \underbrace{0\ldots0}_{m_n}. \quad (15.32)$$

f is a bijection: every order of m balls comes from exactly one bit sequence. Therefore,

$$|B| = |A| = \binom{m+n-1}{m}. \quad (15.33)$$

□

15.4 Sequences with Repetitions

15.4.1 Subset Split Rule

DEFINITION 15.20 **$(k_1, k_2, \ldots, k_r)$-split of a Set** Let S be an n-element set, $\forall i.\ k_i \in \mathbb{N}$, and $\sum_{i=1}^{r} k_i = n$ A $(k_1, k_2, \ldots, k_r)$-split of S is a sequence $(S_1, S_2, \ldots S_r)$ where S_i's are disjoint subsets of S and $\forall i.\ |S_i| = k_i$.

THEOREM 15.21 **Subset Split Rule** The number of $(k_1, k_2, \ldots, k_r)$-split of an n-element set is

$$\binom{n}{k_1, k_2, \ldots, k_r} := \frac{n!}{\prod_{i=1}^{r} k_i!}. \quad (15.34)$$

$\binom{n}{k_1,k_2,\ldots,k_r}$ is called **Multinomial Coefficient**.

Proof. Suppose $S = \{a_1, a_2, \ldots, a_n\}$. Let A be the set of all permutations of the n-element set S. Let B be the set of all possible $(k_1, k_2, \ldots, k_r)$-split of S. Let f be the mapping that takes the first k_1 elements of the permutation from A to the first subset in the split, the second k_2 elements of the permutation from A to the second subset in the split, ..., and the final k_r elements of the permutation from A to the r-th subset of the split.

The mapping f is $k_1!k_2!\cdots k_r!$!-to-1. Therefore

$$|B| = \binom{n}{k_1, k_2, \ldots, k_r} = \frac{|A|}{\prod_{i=1}^{r} k_i!} = \frac{n!}{\prod_{i=1}^{r} k_i!}. \quad (15.35)$$

□

15.4.2 The Bookkeeper Rule

The Bookkeeper Rule is sometimes also called "the formula for permutations with indistinguishable objects".

THEOREM 15.22 **Bookkeeper Rule** Let $c_1, c_2, \ldots, c_r$ be distinct elements. The number of sequences with k_1 occurrences of c_1, k_2 occurrences of c_2 ,..., and k_r occurrences of c_r is

$$\binom{\sum_{i=1}^{r} k_i}{k_1, k_2, \ldots, k_r}. \tag{15.36}$$

Proof. Let A be the set of $(k_1, k_2, \ldots, k_r)$-split of $\{1, 2, \ldots, \sum_{i=1}^{r} k_i\}$. Let B be the set of all sequences with k_1 occurrences of c_1, k_2 occurrences of c_2 ,..., and k_r occurrences of c_r. Let f maps a permutation to the sequence of sets of positions where each of the different letters occur.

From this bijection and the Subset Split Rule,

$$|B| = |A| = \binom{\sum_{i=1}^{r} k_i}{k_1, k_2, \ldots, k_r}. \tag{15.37}$$

□

15.4.3 The Binomial and Multinomial Theorem

DEFINITION 15.23 **Binomial** A sum of two terms.

THEOREM 15.24 **Binomial Theorem**

$$\forall n \in \mathbb{N}, \forall a, b \in \mathbb{R}.\ (a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^{n-k} b^k. \tag{15.38}$$

Proof. By repeatedly using distributivity of products over sums to multiply out this n-th power expression completely, we get a bunch of terms. Each term represents one n-bit sequence of a's and b's, so there are 2^n terms. The number of terms with k copies of b and $n-k$ copies of a is

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \tag{15.39}$$

by the Bookkeeper Rule. Hence the coefficient of $a^{n-k}b^k$ is $\binom{n}{k}$. □

DEFINITION 15.25 **Multinomial** A sum of two or more terms.

THEOREM 15.26 **Multinomial Theorem**

$$\forall n \in \mathbb{N}, \forall a_i \in \mathbb{R}. \left(\sum_{i=1}^{r} a_i\right)^n = \sum_{\substack{k_1,k_2,\ldots,k_r \in \mathbb{N} \\ k_1+k_2+\cdots+k_r=n}} \binom{n}{k_1, k_2, \ldots, k_r} \prod_{i=1}^{r} a_i^{k_i}. \quad (15.40)$$

Proof. By a similar reasoning of the Binomial Theorem. □

15.4.4 Binomial Bounds

THEOREM 15.27

$$\left(\frac{n}{k}\right)^k \leq \binom{n}{k} \leq \left(\frac{en}{k}\right)^k. \quad (15.41)$$

Proof.

$$\binom{n}{k} = \frac{n(n-1)\cdots(n-k+1)}{k(k-1)\cdots 1} = \prod_{i=0}^{k-1} \frac{n-i}{k-i} \geq \prod_{i=0}^{k-1} \frac{n}{k} = \left(\frac{n}{k}\right)^n. \quad (15.42)$$

$$\binom{n}{k} = \frac{1}{k!} \prod_{i=0}^{k-1} (n-i) \leq \frac{1}{k!} \prod_{i=0}^{k-1} n = \frac{n^k}{k!} \leq \left(\frac{en}{k}\right)^k, \quad (15.43)$$

where the last step is by Stirling's Formula $k! \geq (\frac{k}{e})^n$. □

LEMMA 15.28

$$\binom{n}{k} \leq \frac{n^n}{k^k (n-k)^{n-k}}. \quad (15.44)$$

Proof. By induction on k. □

THEOREM 15.29 For $0 \leq \alpha \leq 1$,

$$\binom{n}{\alpha n} \leq 2^{nH(\alpha)}, \quad (15.45)$$

where

$$H(\alpha) = -\alpha \lg \frac{1}{\alpha} - (1-\alpha) \lg \frac{1}{1-\alpha} \quad (15.46)$$

is the **entropy** of α.

Proof.

$$\binom{n}{\alpha n} \leq \frac{n^n}{(\alpha n)^{\alpha n}((1-\alpha)n)^{(1-\alpha)n}} = \left(\left(\frac{1}{\alpha}\right)^{\alpha}\left(\frac{1}{1-\alpha}\right)^{1-\alpha}\right)^n = 2^{nH(n)}. \quad (15.47)$$

□

15.5 Counting Practice: Poker Hands

DEFINITION 15.30 **Hand** 5 cards from a deck of 52 cards. A deck has four kinds of suits and 13 kinds of ranks.

LEMMA 15.31 The number of different possible hands is 2,598,960.

Proof. The number of different possible hands equals to the number of 5-element subsets of a 52-element set, which is

$$\binom{52}{5} = 2,598,960. \quad (15.48)$$

□

15.5.1 Hands with a Four-of-a-Kind

DEFINITION 15.32 **Four-of-a-Kind** A set of four cards with the same rank.

THEOREM 15.33 The number of different possible hands containing a Four-of-a-Kind is 624.

Proof. By mapping this question to a sequence-counting problem

1. Select the rank of the four cards: 13 ways.
2. Select the rank of the extra card: 12 ways.
3. Select the suit of the extra card: 4 ways.

There is a bijection between hands with a Four-of-a-Kind and sequences consisting of two distinct ranks followed by a suit. By the Generalized Product Rule, there are $13 \times 12 \times 4 = 624$ hands with a Four-of-a-Kind. □

15.5.2 Hands with a Full House

DEFINITION 15.34 **Full House** A hand with three cards of one rank and two cards of another rank.

THEOREM 15.35 The number of different possible hands containing a Full House is 3,744.

Proof. By mapping this question to a sequence-counting problem

1. Select the rank of the triple: 13 ways.
2. Select the suits of the triple: $\binom{4}{3}$ ways.
3. Select the rank of the pair: 12 ways.
4. Select the suits of the pair: $\binom{4}{2}$ ways.

There is a bijection between hands with a Full House and the above sequences. By the Generalized Product Rule, there are $13 \times \binom{4}{3} \times 12 \times \binom{4}{2} = 3,744$ hands with a Full House. □

15.5.3 Hands with Two Pairs

DEFINITION 15.36 **Two Pairs** Two cards of one rank, two cards of another rank, and one card of a third rank.

THEOREM 15.37 The number of different possible hands containing Two Pairs is 123,552.

Proof. We proof it using two methods.
Method 1 By mapping this question to a sequence-counting problem

1. Select the rank of the first pair: 13 ways.
2. Select the suits of the first pair: $\binom{4}{2}$ ways.
3. Select the rank of the second pair: 12 ways.
4. Select the suits of the second pair: $\binom{4}{2}$ ways.
5. Select the rank of the extra pair: 11 ways.
6. Select the suits of the extra pair: $\binom{4}{1}$ ways.

Since there is nothing distinguishing the first pair from the second, there is a 2-to-1 mapping between hands with Two Pairs and the above sequences. By the Generalized Product Rule and Division Rule, there are $\frac{13\times\binom{4}{2}\times12\times\binom{4}{2}\times11\times\binom{4}{1}}{2} = 123,552$ hands with Two Pairs.
Method 2 By mapping this question to a sequence-counting problem

1. Select the ranks for the two pairs: $\binom{13}{2}$ ways.
2. Select the suits for the lower-rank pair: $\binom{4}{2}$ ways.
3. Select the suits for the higher-rank pair: $\binom{4}{2}$ ways.
4. Select the rank of the extra pair: 11 ways.
5. Select the suits of the extra pair: $\binom{4}{1}$ ways.

There is a bijection between hands with Two Pairs and the above sequences. By the Generalized Product Rule, there are $\binom{13}{2} \times \binom{4}{2} \times \binom{4}{2} \times 11 \times 4 = 123,552$ hands with Two Pairs. □

15.5.4 Hands with Every Suit

THEOREM 15.38 The number of different possible hands containing at least one card from every suit is 685,464.

Proof. By mapping this question to a sequence-counting problem

1. Select the ranks for each suit: 13^4 ways.
2. Select the suit of the extra card: 4 ways.
3. Select the rank of the extra card: 12 ways.

Since there is nothing distinguishing the two cards with the same rank, there is a 2-to-1 mapping between hands with at least one card from every suit and the above sequences. By the Generalized Product Rule and Division Rule, there are $\frac{13^4 \times 4 \times 12}{2} = 685,464$ hands with at least one card from every suit. □

15.6 The Pigeonhole Principle

15.6.1 The Pigeonhole Principle

THEOREM 15.39 **Pigeonhole Principle** If $|A| > |B|$, then for every total function f $A \mapsto B$, there exist two different elements of set A (the pigeons) that are mapped by f (the rule for assigning pigeons to pigeonholes) to the same element of B (the pigeonholes).

Proof. We prove the contrapositive. Assume for every total function f $A \mapsto B$, different elements of set A are mapped by f to the different elements of B. Therefore $|A| \leq |B|$. □

THEOREM 15.40 **Generalized Pigeonhole Principle** If $|A| > k|B|$, then for every total function f $A \mapsto B$, there exist $(k+1)$ different elements of set A (the pigeons) that are mapped by f (the rule for assigning pigeons to pigeonholes) to the same element of B (the pigeonholes).

Proof. We prove the contrapositive. Assume for every total function f $A \mapsto B$, for every $(k+1)$ different elements of set A are mapped by f to the different elements of B. That is to say, for each element of B, there are at most k elements of A mapped to it. Therefore $|A| \leq k|B|$. □

15.6.2 Hairs on Heads

If you pick two people at random, surely they are extremely unlikely to have exactly the same number of hairs on their heads. However, there is a group of three people who have exactly the same number of hairs out of 500 k non-bald people. The number of hairs on a person's head is at most 200 k.

Let A be the set of non-bald people, $|A| = 5 \times 10^5$. Let $B = \{1, 2, \ldots, 2 \times 10^5\}$ be the number of hairs on a person's head, $|B| = 2 \times 10^5$. Let f map a person to the number of hairs on his or her head. Since $|A| > 2|B|$, the Generalized Pigeonhole Principle implies that at least 3 people have exactly the same number of hairs.

This proof gives no indication who they are, but we know they exist. This frustrating variety of argument is called a **nonconstructive proof**.

15.6.3 Subsets with the Same Sum

Given 10 2-digit numbers, are there two different subsets of these 2-digit numbers that have the same sum?

Let A be the collection of all subsets of the 10 numbers in the list, $|A| = 2^{10} = 1024$ since an n-element set has 2^n different subsets. Let $B = \{0, 1, 2, \ldots, 10 \times 99\}$ be the possible sum of a set of numbers since there are at most 10 numbers in a subset and every 2-digit number is at most 99, $|B| = 991$. Let f map each subset of numbers (in A) to its sum (in B). Since $|A| > |B|$, the Pigeonhole Principle implies that at least 2 different subsets have the same sum.

15.6.4 A Magic Trick

A magician sends an assistant into the audience with a deck of 52 cards while the magician looks away. 5 audience members each select one card from the deck. The assistant then gathers up the 5 cards and holds up 4 of them so the magician can see them. Can the magician determine what the card is?

The trick is: the assistant can choose which of the 5 cards to keep hidden. The problem facing the magician and assistant is actually a bipartite matching problem. Let V_L denotes the set of vertices on the left which will correspond to the information available to the assistant, namely, a set of 5 cards, so

$$|V_L| = \binom{52}{5} = 2,598,960\,. \tag{15.49}$$

Let V_R denotes the set of vertices on the right which will correspond to the information available to the magician, namely, a sequence of 4 distinct cards, so

$$|V_R| = 52 \times 51 \times 50 \times 49 = 6,497,400\,. \tag{15.50}$$

When the audience selects a set of 5 cards, then the assistant must reveal a sequence of 4 cards from that hand. This constraint is represented by having an edge between a set of 5 cards on the left and a sequence of 4 cards on the right precisely when every card in the sequence is also in the set. This specifies the bipartite graph.

What the magician and his assistant need to perform the trick is a matching between V_L and V_R. If they agree in advance on some matching, then when the audience selects a set of 5 cards, the Assistant reveals the matching sequence of 4 cards. The Magician uses the matching to find the audience's chosen set of 5 cards, and so he can name the one not already revealed.

LEMMA 15.41

$$\forall u \in V_L.\ \deg u = 120\,. \tag{15.51}$$

Proof. Given a set of 5 cards, there are five ways to select the card kept secret and 4! permutations of the remaining 4 cards, in total, $5 \times 4! = 120$. □

LEMMA 15.42

$$\forall v \in V_R.\ \deg v = 48\,. \tag{15.52}$$

Proof. Given a sequence of 4 cards, there are 48 possibilities for the 5-th card. □

THEOREM 15.43 There is a matching between V_L and V_R.

Proof. By the above two lemmas, the graph is degree-constrained. Therefore, it has a matching. □

15.6.5 The Real Secret

It is all very well in principle to have the magician and his assistant agree on a matching, but how are they supposed to remember a matching with $|V_L| = 2,598,960$ edges? For the trick to work in practice, there has to be a way to match hands and card sequences mentally and on the fly.

1. After the audience select 5 cards, the assistant picks out two cards of the same suit. This is always possible because of the Pigeonhole Principle, there are five cards and 4 suits so two cards must be in the same suit.
2. The assistant locates the ranks of these two cards on the cycle. For any two distinct ranks on this cycle, one is always between 1 and 6 hops clockwise from the other.

3. The more counterclockwise of these two cards is revealed first, and the other becomes the secret card.
4. The magician and assistant agree beforehand on an ordering of all the cards in the deck from smallest to largest. The order in which the last three cards are revealed communicates the number to reach the secret card.

15.7 Inclusion-Exclusion Principle

15.7.1 Union of Two Sets

THEOREM 15.44 For two sets A and B,

$$|A \cup B| = |A| + |B| - |A \cap B| . \tag{15.53}$$

Proof.

$$A \cup B = A \cup (B - A) . \tag{15.54}$$

Because A and $B - A$ are disjoint, we can apply the sum rule

$$|A \cup B| = |A| + |B - A| . \tag{15.55}$$

□

COROLLARY 15.45 For two sets A and B,

$$|A \cup B| \leq |A| + |B| . \tag{15.56}$$

The equality holds when A and B are disjoint, i.e., $A \cap B = \varnothing$.

15.7.2 Union of Three Sets

THEOREM 15.46 For three sets A, B, and C,

$$\begin{aligned} |A \cup B \cup C| = {} & |A| + |B| + |C| \\ & -|A \cap B| - |A \cap C| - |B \cap C| \\ & +|A \cap B \cap C| . \end{aligned} \tag{15.57}$$

Proof. By a similar argument as the two sets case. □

15.7.3 Union of n Sets

THEOREM 15.47 For n sets $S_1, S_2, \ldots, S_n$,

$$\left|\bigcup_{i=1}^{n} S_i\right| = \sum_{i=1}^{n} |S_i| - \sum_{1\le i<j\le n} |S_i \cap S_j| + \sum_{1\le i<j<k\le n} |S_i \cap S_j \cap S_k| + \cdots + (-1)^{n-1} \left|\bigcap_{i=1}^{n} S_i\right| \tag{15.58}$$

$$= \sum_{A\subseteq\{1,2,\ldots,n\}, A\neq\varnothing} (-1)^{|A|+1} \left|\bigcap_{i\in A} S_i\right| . \tag{15.59}$$

Proof. By induction on n. □

15.7.4 Sequences with 42, 04, or 60

THEOREM 15.48 There are 972,720 permutations of the set $\{0, 1, 2, \ldots, 9\}$ do either $(4, 2)$, $(0, 4)$, or $(6, 0)$ appear consecutively.

Proof. Let S_{42} be the set of all permutations in which $(4, 2)$ appears. Let S_{60} be the set of all permutations in which $(6, 0)$ appears. Let S_{04} be the set of all permutations in which $(0, 4)$ appears.

First, we must determine the sizes of the individual sets, such as S_{60}. We can group 6 and 0 together as a single symbol. Then there is an immediate bijection between permutations of $\{0, 1, 2, \ldots, 9\}$ containing $(6, 0)$ consecutively and permutations of $\{60, 1, 2, 3, 4, 5, 7, 8, 9\}$. There are 9! permutations of the set containing $(6, 0)$, so $|S_{60}| = 9!$. Similarly, $|S_{04}| = |S_{42}| = 9!$ as well.

Next, we must determine the sizes of the two-way intersections, such as $S_{42} \cap S_{60}$. Using the grouping trick again, there is a bijection with permutations of the set $\{42, 60, 1, 3, 5, 7, 8, 9\}$. Thus, $|S_{42} \cap S_{60}| = 8!$. Similarly, $|S_{42} \cap S_{60}| = 8!$ by a bijection with the set $\{604, 1, 2, 3, 5, 7, 8, 9\}$, and $|S_{42} \cap S_{04}| = 8!$ as well by a similar argument.

Finally $|S_{42} \cap S_{04} \cap S_{60}| = 7!$ by a bijection with the set $\{6042, 1, 2, 3, 5, 7, 8, 9\}$.

Plugging all this into the formula gives

$$|S_{42} \cap S_{04} \cap S_{60}| = 9! + 9! + 9! - 8! - 8! - 8! + 7! = 972,720 . \tag{15.60}$$

□

15.7.5 Computing Euler's Totient Function

THEOREM 15.49 If the prime factorization of n is $p_1^{e_1}p_2^{e_2}\cdots p_k^{e_k}$ for distinct primes p_i, then

$$\phi(n) = n\prod_{i=1}^{k}\left(1-\frac{1}{p_i}\right). \tag{15.61}$$

Proof. Let

$$S = \{i \in [0,n) \mid \gcd(n,i) \neq 1\} \tag{15.62}$$

be the set of integers in $[0,n)$ that are not relatively prime to n. So $\phi(n) = n - |S|$.

Let

$$C_d = \{i \in [0,n) : d \mid i\} \tag{15.63}$$

be the set of integers in $[0,n)$ that are divisible by d. So the integers in S are precisely the integers in $[0,n)$ that are divisible by at least one of the p_i's. Namely,

$$S = \bigcup_{i=1}^{k} C_{p_i}. \tag{15.64}$$

The intersections of the C_{p_i}'s are easy to count. For example, for 3 distinct primes p_1, p_2, and p_3, $C_{p_1} \cap C_{p_2} \cap C_{p_3}$ is the set of integers in $[0,n)$ that are divisible by each of p_1, p_2, and p_3. since being divisible by each of them is the same as being divisible by their product, and if d is a positive divisor of n, then there are exactly $\frac{n}{d}$ multiples of d in $[0,n)$. Therefore,

$$|C_{p_1} \cap C_{p_2} \cap C_{p_3}| = \frac{n}{p_1p_2p_3}. \tag{15.65}$$

This reasoning extends to arbitrary intersections of C_{p_i}'s, namely, for any nonempty set $A \subseteq \{1,2,\ldots,k\}$,

$$\left|\bigcap_{i\in A} C_{p_i}\right| = \frac{n}{\prod_{i\in A} p_i}. \tag{15.66}$$

Therefore

$$|S| = \left|\bigcup_{i=1}^{k} C_{p_i}\right| \tag{15.67}$$

$$= \sum_{A\subseteq\{1,2,\ldots,k\}, A\neq\varnothing} (-1)^{|A|+1}\left|\bigcap_{i\in A} S_i\right| \tag{15.68}$$

$$= \sum_{A\subseteq\{1,2,\ldots,k\},A\neq\varnothing} (-1)^{|A|+1} \frac{n}{\prod_{i\in A} p_i} \tag{15.69}$$

$$= -n \sum_{A\subseteq\{1,2,\ldots,k\},A\neq\varnothing} \frac{1}{\prod_{i\in A}(-p_i)} \tag{15.70}$$

$$= -n\left(\prod_{i=1}^{k}\left(1-\frac{1}{p_i}\right)-1\right) \tag{15.71}$$

$$= -n\prod_{i=1}^{k}\left(1-\frac{1}{p_i}\right)+n\,. \tag{15.72}$$

$$\therefore\ \phi(n) = n-|S| = n\prod_{i=1}^{k}\left(1-\frac{1}{p_i}\right). \tag{15.73}$$

□

15.8 Combinatorial Proofs

15.8.1 Giving a Combinatorial Proof

DEFINITION 15.50 **Combinatorial Proof** An argument that establishes an algebraic fact by relying on counting principles.

Many combinatorial proofs follow the same basic outline

1. Define a set S.
2. Show that $|S| = n$ by counting one way.
3. Show that $|S| = m$ by counting another way.
4. Conclude that $n = m$.

15.8.2 Example

THEOREM 15.51

$$\sum_{k=0}^{n}\binom{n}{k}\binom{2n}{n-k} = \binom{3n}{n}. \tag{15.74}$$

Proof. We give a combinatorial proof. Let S be all n balls that can be drawn from a box containing n different balls and $2n$ different black balls.

First, note that every $3n$-element set has

$$|S| = \binom{3n}{n} \tag{15.75}$$

n-element subsets.

From another perspective, the number of ways to get n balls with exactly k red balls and $n-k$ black balls is

$$\binom{n}{k}\binom{2n}{n-k}. \tag{15.76}$$

Since the number of red balls can be anywhere from 0 to n, the total number of ways is

$$|S| = \sum_{k=0}^{n} \binom{n}{k}\binom{2n}{n-k}. \tag{15.77}$$

Equating these two expressions for $|S|$ proves the theorem. □

15.8.3 Pascal's Triangle Identity

LEMMA 15.52 [Pascal's Triangle Identity]

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}. \tag{15.78}$$

Proof. We give a combinatorial proof. Let S be all k-element subsets of an n-element set A, and $a \in A$.

First, note that every n-element set has

$$|S| = \binom{n}{k} \tag{15.79}$$

k-element subsets.

From another perspective, for counting the total number of k-element subsets, if a is selected for the subset, the number of different k-element subsets that can be formed in this way is $\binom{n-1}{k-1}$; if a is not selected for the subset, the number of different k-element subsets that can be formed in this way is $\binom{n-1}{k}$. Since all subsets of the first type contain a, and no subset of the second type does, the two sets of subsets are disjoint. Thus by the Sum Rule, the total number of k-element subsets is

$$\binom{n-1}{k-1} + \binom{n-1}{k}. \tag{15.80}$$

Equating these two expressions for $|S|$ proves the theorem. □

16 Generating Functions

Generating functions transform problems about *sequences* into problems about *functions*. Generating functions are particularly useful for representing and counting the number of ways to select n things.

16.1 Formal Power Series

16.1.1 The Ring of Power Series

DEFINITION 16.1 **Sequence Sum** $\oplus$ The sequence sum $\oplus$ of two infinite sequences $F := (f_n)_{n=0}^{\infty}$ and $G := (g_n)_{n=0}^{\infty}$ is

$$F \oplus G := (f_n + g_n)_{n=0}^{\infty} = (f_0 + g_0, f_1 + g_1, \ldots, f_n + g_n, \ldots), \tag{16.1}$$

DEFINITION 16.2 **Sequence Multiplication** $\otimes$ The sequence sum $\oplus$ of two infinite sequences $F := (f_n)_{n=0}^{\infty}$ and $G := (g_n)_{n=0}^{\infty}$ is

$$F \otimes G := \left(\sum_{k=0}^{n} f_k g_{n-k} \right)_{n=0}^{\infty} = \left(f_0 + g_0, \sum_{k=0}^{1} f_k g_{1-k}, \ldots, \sum_{k=0}^{n} f_k g_{n-k}, \ldots \right). \tag{16.2}$$

The sequence of coefficients of the product $F \otimes G$ is called the **convolution** of the sequences F and G.

LEMMA 16.3 Sequence sum $\oplus$ and sequence multiplication $\otimes$ are commutative

$$F \oplus G = G \oplus F, \tag{16.3}$$

$$F \otimes G = G \otimes F. \tag{16.4}$$

THEOREM 16.4 Sequence sum $\oplus$ and sequence multiplication $\otimes$ satisfy all the commutative ring axioms.

DEFINITION 16.5 **Formal Power Series** The set of infinite sequences of numbers together with sequence sum $\oplus$ and sequence multiplication $\otimes$ operations.

DEFINITION 16.6 **Sequence Zero** Z

$$Z := (0, 0, \ldots, 0, \ldots). \tag{16.5}$$

DEFINITION 16.7 **Sequence Identity** I

$$I := (1, 0, \ldots, 0, \ldots). \tag{16.6}$$

LEMMA 16.8 Z is the zero element and I is the identity element for sequence sum $\oplus$ and sequence multiplication $\otimes$.

$$Z \oplus F = F, \tag{16.7}$$

$$Z \otimes F = Z, \tag{16.8}$$

$$I \otimes F = F. \tag{16.9}$$

DEFINITION 16.9 **Sequence Negative $-F$** The sequence negative of an infinite sequence $F := (f_n)_{n=0}^{\infty}$ is

$$-F := (-f_n)_{n=0}^{\infty} = (-f_0, -f_1, \ldots, -f_n, \ldots). \tag{16.10}$$

LEMMA 16.10

$$F \oplus -F = Z. \tag{16.11}$$

DEFINITION 16.11 **Sequence Reciprocal/Multiplicative Inverse** A sequence G is the reciprocal of a sequence F when

$$F \otimes G = I. \tag{16.12}$$

LEMMA 16.12 If the sequence reciprocal exists, it is unique.

Proof. It follows by the ring axioms. □

LEMMA 16.13 A series has an inverse iff its initial element is nonzero.

16.1.2 Ordinary Generating Functions

DEFINITION 16.14 **Ordinary Generating Function** The ordinary generating function $F(x)$ for the sequence $(f_i)_{i=0}^{\infty}$ is the power series

$$F(x) := \sum_{n=0}^{\infty} f_n x^n. \tag{16.13}$$

We use the notation $[x^n]\,F(x)$ for the coefficient of x^n in the generating function $F(x)$. That is

$$\left[x^n\right] F(x) := f_n. \tag{16.14}$$

A generating function $F(x)$ really refers to its infinite sequence of coefficients $(f_i)_{i=0}^{\infty}$ in the ring of formal power series. The powers of the variable x just serve as a place holders, and the equation has nothing to do with the values of x or the convergence of the series.

We can analyze the behavior of any sequence of numbers $f_0, f_1, \ldots, f_n$ regarding the elements of the sequence as successive coefficients of a generating function, and we can

carry out all sorts of manipulations on sequences by performing mathematical operations on their associated generating functions.

LEMMA 16.15 The followings are some basic sequences and their generating functions:

$$(1,1,1,1,\ldots) \Leftrightarrow \sum_{n=0}^{\infty} x^n = \frac{1}{1-x}, \tag{16.15}$$

$$(1,-1,1,-1,\ldots) \Leftrightarrow \sum_{n=0}^{\infty} (-1)^n x^n = \frac{1}{1+x}, \tag{16.16}$$

$$(1,a,a^2,a^3,\ldots) \Leftrightarrow \sum_{n=0}^{\infty} a^n x^n = \frac{1}{1-ax}, \tag{16.17}$$

$$(1,0,1,0,\ldots) \Leftrightarrow \sum_{n=0}^{\infty} (x^2)^n = \frac{1}{1-x^2}. \tag{16.18}$$

Proof. By perturbation methods. □

16.2 Operations with Generating Functions

THEOREM 16.16 **Scaling Rule** If

$$(f_n)_{n=0}^{\infty} \Leftrightarrow F(x), \tag{16.19}$$

then

$$(cf_n)_{n=0}^{\infty} \Leftrightarrow cF(x). \tag{16.20}$$

Proof.

$$(cf_n)_{n=0}^{\infty} \Leftrightarrow \sum_{n=0}^{\infty} cf_n x^n = c\sum_{n=0}^{\infty} f_n x^n = cF(x). \tag{16.21}$$

□

THEOREM 16.17 **Addition Rule** If

$$(f_n)_{n=0}^{\infty} \Leftrightarrow F(x) \wedge (g_n)_{n=0}^{\infty} \Leftrightarrow G(x), \tag{16.22}$$

then

$$(f_n)_{n=0}^{\infty} \oplus (g_n)_{n=0}^{\infty} \Leftrightarrow F(x) + G(x). \tag{16.23}$$

Proof.

$$(f_n)_{n=0}^{\infty} \oplus (g_n)_{n=0}^{\infty} \Leftrightarrow (f_n + g_n)_{n=1}^{\infty} \Leftrightarrow \sum_{n=0}^{\infty} (f_n + g_n)x^n = \sum_{n=0}^{\infty} f_n x^n + \sum_{n=0}^{\infty} g_n x^n = F(x) + G(x) \tag{16.24}$$

□

THEOREM 16.18 **Right Shifting Rule** If

$$(f_n)_{n=0}^{\infty} \Leftrightarrow F(x)\,, \tag{16.25}$$

then

$$(\underbrace{0, 0, \ldots, 0}_{k \text{ zeros}}, f_0, f_1, \ldots) \Leftrightarrow x^k F(x)\,. \tag{16.26}$$

Proof.

$$(\underbrace{0, 0, \ldots, 0}_{k \text{ zeros}}, f_0, f_1, \ldots) \Leftrightarrow \sum_{n=0}^{\infty} f_n x^{k+n} = x^k \sum_{n=0}^{\infty} f_n x^n = x^k F(x)\,. \tag{16.27}$$

□

THEOREM 16.19 **Differentiation Rule** If

$$(f_n)_{n=0}^{\infty} \Leftrightarrow F(x)\,, \tag{16.28}$$

then

$$(n f_n)_{n=1}^{\infty} \Leftrightarrow \frac{\mathrm{d}}{\mathrm{d}x} F(x)\,. \tag{16.29}$$

That is to say, each term is multiplied by its index and the entire sequence is shifted left by one place.

Proof.

$$(n f_n)_{n=1}^{\infty} = ((n+1) f_{n+1})_{n=0}^{\infty}\,, \tag{16.30}$$

which corresponds to

$$\sum_{n=0}^{\infty} (n+1) f_{n+1} x^n = \frac{\mathrm{d}}{\mathrm{d}x} \sum_{n=0}^{\infty} f_{n+1} x^{n+1} = \frac{\mathrm{d}}{\mathrm{d}x} \sum_{n=1}^{\infty} f_n x^n = \frac{\mathrm{d}}{\mathrm{d}x} \left(f_0 + \sum_{n=1}^{\infty} f_n x^n \right) = \frac{\mathrm{d}}{\mathrm{d}x} F(x) \tag{16.31}$$

□

COROLLARY 16.20 **Index Multiplication Rule** If

$$(f_n)_{n=0}^{\infty} \Leftrightarrow F(x)\,, \tag{16.32}$$

then

$$(nf_n)_{n=0}^{\infty} \Leftrightarrow x\frac{\mathrm{d}}{\mathrm{d}x}F(x)\,. \tag{16.33}$$

That is to say, each term is multiplied by its index.

Proof. By Differentiation Rule and Right Shifting Rule,

$$(nf_n)_{n=1}^{\infty} \Leftrightarrow \frac{\mathrm{d}}{\mathrm{d}x}F(x)\,, \tag{16.34}$$

$$(0, f_1, 2f_2, \ldots) = (nf_n)_{n=0}^{\infty} \Leftrightarrow x\frac{\mathrm{d}}{\mathrm{d}x}F(x)\,. \tag{16.35}$$

□

THEOREM 16.21 **Product Rule** If

$$(f_n)_{n=0}^{\infty} \Leftrightarrow F(x) \wedge (g_n)_{n=0}^{\infty} \Leftrightarrow G(x)\,, \tag{16.36}$$

then

$$(f_n)_{n=0}^{\infty} \otimes (g_n)_{n=0}^{\infty} \Leftrightarrow F(x) \cdot G(x)\,. \tag{16.37}$$

Proof.

$$(f_n)_{n=0}^{\infty} \otimes (g_n)_{n=0}^{\infty} = \left(\sum_{k=0}^{n} f_k g_{n-k}\right)_{n=0}^{\infty} = \sum_{n=0}^{\infty}\left(\sum_{k=0}^{n} f_k g_{n-k}\right)x^n = F(x) \cdot G(x) \tag{16.38}$$

□

COROLLARY 16.22 **Summation Rule** If

$$(f_n)_{n=0}^{\infty} \Leftrightarrow F(x)\,, \tag{16.39}$$

then

$$\left(\sum_{k=0}^{n} f_k\right)_{n=0}^{\infty} \Leftrightarrow \frac{F(x)}{1-x}\,. \tag{16.40}$$

Proof. Let

$$(1)_{n=0}^{\infty} \Leftrightarrow G(x) := \sum_{n=0}^{\infty} x^n = \frac{1}{1-x}. \tag{16.41}$$

By Product Rule,

$$(f_n)_{n=0}^{\infty} \otimes (1)_{n=0}^{\infty} = \left(\sum_{k=0}^{n} f_k \cdot 1 \right)_{n=0}^{\infty} \Leftrightarrow F(x)G(x) = \frac{F(x)}{1-x}. \tag{16.42}$$

□

16.3 Extract Coefficients

16.3.1 Maclaurin's Theorem

THEOREM 16.23 **Maclaurin's Theorem**

$$f(x) = \sum_{n=0}^{\infty} \frac{1}{n!} \frac{\mathrm{d}^n f(0)}{\mathrm{d}x^n} x^n. \tag{16.43}$$

That is to say, for a generation function $F(x)$,

$$\left[x^n\right] F(n) = \frac{1}{n!} \frac{\mathrm{d}^n F(0)}{\mathrm{d}x^n}. \tag{16.44}$$

COROLLARY 16.24 The followings are some well-known series

$$\frac{1}{1-x} = \sum_{n=0}^{\infty} x^n, \tag{16.45}$$

$$\exp x = \sum_{n=0}^{\infty} \frac{1}{n!} x^n, \tag{16.46}$$

$$\exp ax = \sum_{n=0}^{\infty} \frac{a^n}{n!} x^n, \tag{16.47}$$

$$\log(1-x) = -\sum_{n=1}^{\infty} \frac{a^n}{n} x^n. \tag{16.48}$$

LEMMA 16.25

$$\forall k \in \mathbb{N}^+. \ \frac{\mathrm{d}^n}{\mathrm{d}x^n} \frac{1}{(1-\alpha x)^k} = \frac{(n+k-1)!\alpha^n}{(k-1)!(1-\alpha x)^{k+n}}. \tag{16.49}$$

Proof. By induction on n. Let the inductive hypothesis $P(n)$ be the claim of the lemma.
Base case: $n = 1$,

$$\frac{\mathrm{d}}{\mathrm{d}x}\frac{1}{(1-\alpha x)^k} = \frac{k\alpha}{(1-\alpha x)^{k+1}} = \frac{(1+k-1)!\alpha^1}{(k-1)!(1-\alpha x)^{k+1}} \tag{16.50}$$

Inductive step: We assume the $P(n)$ is true,

$$\frac{\mathrm{d}^{n+1}}{\mathrm{d}x^{n+1}}\frac{1}{(1-\alpha x)^k} = \frac{\mathrm{d}}{\mathrm{d}x}\frac{(n+k-1)!\alpha^n}{(k-1)!(1-\alpha x)^{k+n}} = \frac{(n+k-1)!\alpha^n(k+n)\alpha}{(k-1)!(1-\alpha x)^{k+n+1}} = \frac{(n+1+k-1)!\alpha^{n+1}}{(k-1)!(1-\alpha x)^{k+n+1}} \tag{16.51}$$

□

16.3.2 Partial Fractions

Maclaurin's Theorem is a very general method for finding coefficients, but it only applies when formulas for repeated derivatives can be found, which is not often. However, there is an automatic way to find the power series coefficients for any formula that is a quotient of polynomials, namely, the method of partial fractions from elementary calculus.

16.3.2.1 Partial Fractions Without Repeated Roots

LEMMA 16.26 Let $p(x)$ be a polynomial of degree less than n and let $\alpha_1, \alpha_2, \ldots, \alpha_n$ be distinct, nonzero numbers. Then there are constants $c_1, c_2, \ldots, c_n$ such that

$$\frac{p(x)}{\prod_{i=1}^{n}(1-\alpha_i x)} = \sum_{i=1}^{n}\frac{c_i}{1-\alpha_i x}\,. \tag{16.52}$$

c_i can be determined by multiplying both sides by the left-hand denominator to get

$$p(x) = \sum_{i=1}^{n}\frac{c_i}{1-\alpha_i x}\prod_{j=1}^{n}(1-\alpha_j x)\,, \tag{16.53}$$

and then let $x = \frac{1}{\alpha_i}$ to get

$$c_i = p\left(\frac{1}{\alpha_i}\right) \Big/ \prod_{\substack{1\le j\le n\\ j\neq i}}(1-\alpha_j x)\,. \tag{16.54}$$

Each term in the partial fractions expansion has a simple power series given by the geometric sum formula:

$$\frac{c_i}{1-\alpha_i x} = c_i\sum_{n=0}^{\infty}(\alpha_i x)^n = c_i\sum_{n=0}^{\infty}\alpha_i^n x^n\,. \tag{16.55}$$

Then

$$\left[x^n\right]\left(\frac{c_i}{1-\alpha_i x}\right) = c_i\alpha_i^n\,. \tag{16.56}$$

16.3.2.2 Partial Fractions with Repeated Roots

When the denominator polynomial has a repeated nonzero root $\frac{1}{\alpha_i} \neq 0$ with multiplicity m, we expand the quotient into a sum a terms of the form

$$\sum_{k=1}^{m} \frac{c_i}{(1-\alpha_i x)^k}\,. \tag{16.57}$$

A formula for the coefficients of such a term follows from the donut formula

$$[x^n]\left(\frac{c_i}{(1-\alpha_i x)^k}\right) = c_i \binom{n+k-1}{n} \alpha_i^n\,. \tag{16.58}$$

16.4 Counting with Generating Functions

Problems involving choosing items from a set often lead to nice generating functions by letting the coefficient of x^n be the number of ways to choose n items.

16.4.1 Convolution Rule

LEMMA 16.27 Suppose we have two kinds of things, say red balls and green balls. Say there are a_n ways to select n red balls and b_n ways to select n green balls. This means that the total number of ways to select some size n mix of red and green balls is

$$\sum_{k=0}^{n} a_k b_{n-k}\,, \tag{16.59}$$

Proof. The number of red balls to select can be any number k from 0 to n. We can then select these red balls in a_k ways, by definition. This leaves $n-k$ green balls to be selected, which be definition can be done in b_{n-k} ways. So the total number of ways to select k red balls and $n-k$ green balls is $a_k b_{n-k}$. This means that the total number of ways to select some size n mix of red and green balls is $\sum_{k=0}^{n} a_k b_{n-k}$. □

THEOREM 16.28 **Convolution Rule** Let $F(x)$ be the generating function for selection items from a set A, and $G(x)$ be the generating function for selection items from a set B, and A and B are disjoint. The generating function for selecting items from the union $A \cup B$ is the product $F(x) \cdot G(x)$. In other words, the generating function for choosing elements from a union of disjoint sets is the product of the generating functions for choosing from each set.

16.4.2 Choosing Items with Repetition

THEOREM 16.29 Suppose there are k flavors of donuts, The number of ways to select n donuts is $\binom{n+k-1}{n}$.

Proof. Let $A_i(x)$ be the generating function for selecting donuts of the k-the favor, Since there is only one way to select exactly n donuts of the k-the favor,

$$A_i(x) = \sum_{n=0}^{\infty} x^n = \frac{1}{1-x}, \forall i. \tag{16.60}$$

By the Convolution Rule, the generating function for the number of ways to select n donuts when both chocolate and plain flavors are available is

$$G(x) = \prod_{i=1}^{k} A_i(x) = \prod_{i=1}^{k} \frac{1}{1-x} = \frac{1}{(1-x)^k}. \tag{16.61}$$

By Maclaurin's Theorem, the number of ways to select n donuts is

$$\begin{aligned}
\left[x^n\right] G(x) &= \left[x^n\right] \left(\frac{1}{(1-x)^k}\right) && (16.62)\\
&= \frac{1}{n!} \frac{\mathrm{d}^n}{\mathrm{d}x^n} \left(\frac{1}{(1-x)^k}\right)\Big|_{x=0} && (16.63)\\
&= \frac{1}{n!} k(k+1)\cdots(k+n-1)(1-x)^{-(k+n)}\Big|_{x=0} && (16.64)\\
&= \frac{(n+k-1)!}{n!(k-1)!} && (16.65)\\
&= \binom{n+k-1}{n}. && (16.66)
\end{aligned}$$

□

16.4.3 The Binomial Theorem

THEOREM 16.30 The number of ways to select k elements from a set of size n is $\binom{n}{k}$.

Proof. Let $A_i(x)$ be the generating function for selecting elements from the single-element set $\{a_i\}$, since there is 1 way to select zero element, 1 way to select one element, and 0 ways to select more than one element,

$$A_i(x) = 1 + x, \forall i. \tag{16.67}$$

By the Convolution Rule, the generating function for choosing a subset of k elements from the set $\{a_1, a_2, \ldots, a_n\}$ is

$$G(x) = \prod_{i=1}^{n} A_i(x) = (1+x)^n. \tag{16.68}$$

By Maclaurin's Theorem, the number of ways to select k elements is

$$\left[x^k\right] G(x) = \left[x^k\right]\left((1+x)^n\right) \tag{16.69}$$

$$= \frac{1}{k!}\frac{\mathrm{d}^k}{\mathrm{d}x^k}\left((1+x)^n\right)\Big|_{x=0} \tag{16.70}$$

$$= \frac{1}{k!}n(n-1)\dots(n-k+1)(1+x)^{n-k}\Big|_{x=0} \tag{16.71}$$

$$= \frac{n!}{k!(n-k)!} \tag{16.72}$$

$$= \binom{n}{k}. \tag{16.73}$$

□

16.4.4 An Absurd Counting Problem

THEOREM 16.31 There is $n+1$ ways to fill a bag with n fruits subject to the following constraints

- The number of apples must be even.
- The number of bananas must be a multiple of 5.
- There can be at most four oranges.
- There can be at most one pear.

Proof. Let $A(x)$ be the generating function for selecting apples, so

$$A(x) = 1 + x^2 + x^4 + x^6 + \cdot = \sum_{n=0}^{\infty}(x^2)^n = \frac{1}{1-x^2}. \tag{16.74}$$

Let $B(x)$ be the generating function for selecting bananas, so

$$B(x) = 1 + x^5 + x^{10} + x^{15} + \cdots = \sum_{n=0}^{\infty}(x^5)^n = \frac{1}{1-x^5}. \tag{16.75}$$

Let $C(x)$ be the generating function for selecting oranges, so

$$C(x) = 1 + x + x^2 + x^3 + x^4 = \frac{1-x^5}{1-x}. \tag{16.76}$$

Let $D(x)$ be the generating function for selecting pears, so

$$D(x) = 1 + x. \tag{16.77}$$

By the Convolution Rule, the generating function for choosing n fruits from all four kinds is

$$G(x) = A(x)B(x)C(x)D(x) = \frac{1}{1-x^2}\frac{1}{1-x^5}\frac{1-x^5}{1-x}(1+x) = \frac{1}{(1-x)^2} = \sum_{n=p}^{\infty}(n+1)x^n \tag{16.78}$$

So the number of ways to fill a bag with n fruits is $n+1$. □

16.5 Solving Linear Recurrences

16.5.1 Fibonacci Numbers

DEFINITION 16.32 **Fibonacci Numbers** The n-th Fibonacci number is defined recursively as follows

$$f_n := \begin{cases} 0 & \text{if } n = 0; \\ 1 & \text{if } n = 1; \\ f_{n-1} + f_{n-2} & \text{if } n \geq 2. \end{cases} \tag{16.79}$$

THEOREM 16.33 **Binet's Formula** The n-th Fibonacci number f_n is

$$f_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n. \tag{16.80}$$

Proof. Let $F(x)$ be the generating function for the sequence of Fibonacci numbers

$$F(x) = \sum_{n=0}^{\infty} f_n x^n. \tag{16.81}$$

We use the perturbation method, namely,

$$-xF(x) = \sum_{n=0}^{\infty}(-f_n)x^{n+1} = \sum_{n=1}^{\infty}(-f_{n-1})x^n, \tag{16.82}$$

$$-x^2F(x) = \sum_{n=0}^{\infty}(-f_n)x^{n+2} = \sum_{n=2}^{\infty}(-f_{n-2})x^n. \tag{16.83}$$

Therefore

$$F(x)(1-x-x^2) = f_0 + (f_1 - f_0)x + \sum_{n=2}^{\infty}(f_n - f_{n+1} - f_{n+2})x^n = x. \tag{16.84}$$

$$\therefore F(x) = \frac{x}{1-x-x^2} = \frac{x}{(1-\alpha_1 x)(1-\alpha_2 x)} = \frac{c_1}{1-\alpha_1 x} + \frac{c_2}{1-\alpha_2 x}. \tag{16.85}$$

where

$$\alpha_1 = \frac{1+\sqrt{5}}{2}, \tag{16.86}$$

$$\alpha_2 = \frac{1-\sqrt{5}}{2} \tag{16.87}$$

are the reciprocals of the roots $1 - x - x^2 = 0$. c_1 and c_2 can be computed by

$$c_1 = \frac{\frac{1}{\alpha_1}}{1-\frac{\alpha_2}{\alpha_1}} = \frac{1}{\sqrt{5}}, \tag{16.88}$$

$$c_2 = \frac{\frac{1}{\alpha_2}}{1-\frac{\alpha_1}{\alpha_2}} = -\frac{1}{\sqrt{5}}. \tag{16.89}$$

$$\therefore\ F(x) = \frac{1}{\sqrt{5}}\frac{1}{1-\alpha_1 x} - \frac{1}{\sqrt{5}}\frac{1}{1-\alpha_2 x}. \tag{16.90}$$

$$\left[x^n\right]F(x) = \frac{1}{\sqrt{5}}\alpha_1^n - \frac{1}{\sqrt{5}}\alpha_2^n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n. \tag{16.91}$$

□

16.5.2 The Towers of Hanoi

The recurrence of Hanoi problem is

$$f_n = \begin{cases} 0 & \text{if } n = 0; \\ 2f_{n-1} + 1 & \text{if } n \geq 1. \end{cases} \tag{16.92}$$

THEOREM 16.34 $f_n = 2^n - 1$ for the Hanoi problem.

Proof. Let $F(x)$ be the generating function for the Hanoi problems

$$F(x) = \sum_{n=0}^{\infty} f_n x^n. \tag{16.93}$$

We use the perturbation method, namely,

$$2xF(x) = \sum_{n=0}^{\infty} 2f_n x^{n+1} = \sum_{n=1}^{\infty} 2f_{n-1}x^n. \tag{16.94}$$

So

$$F(x)(1-2x) = f_0 + \sum_{n=1}^{\infty}(f_n - 2f_{n-1})x^n = \sum_{n=1}^{\infty} x^n = \frac{1}{1-x} - 1 = \frac{x}{1-x}. \tag{16.95}$$

$$\therefore\ F(x) = \frac{x}{(1-2x)(1-x)} = \frac{c_1}{1-2x} + \frac{c_2}{1-x}. \tag{16.96}$$

c_1 and c_2 can be computed by

$$c_1 = \frac{\frac{1}{2}}{1-\frac{1}{2}} = 1\,, \tag{16.97}$$

$$c_1 = \frac{1}{1-2} = -1\,. \tag{16.98}$$

$$\therefore\ F(x) = \frac{1}{1-2x} - \frac{1}{1-x}\,. \tag{16.99}$$

$$\therefore \left[x^n\right] F(x) = 2^n - 1\,. \tag{16.100}$$

□

16.5.3 Solving General Linear Recurrences

The methods above can be used to solve linear recurrences with a large class of inhomogeneous terms. In particular, when the inhomogeneous term itself has a generating function that can be expressed as a quotient of polynomials, f_n is a linear combination of terms of the form $n^k\alpha^n$ where $k \le d$ is a nonnegative integer and α is the reciprocal of a root of the denominator polynomial.

16.6 Formal Power Series

16.6.1 Divergent Generating Functions

THEOREM 16.35

$$n! = 1 + \sum_{i=1}^{n} (i-1)\cdot(i-1)!\,. \tag{16.101}$$

Proof. Let $F(x)$ be the generating function for $n!$,

$$F(x) = \sum_{n=0}^{\infty} n! x^n\,. \tag{16.102}$$

Let $H(x)$ be the generating function for $n \cdot n!$,

$$H(x) = \sum_{n=0}^{\infty} n \cdot n! x^n\,. \tag{16.103}$$

By using perturbation method,

$$\frac{F(x)}{x} = \frac{0!}{x} + \sum_{n=1}^{\infty} n! x^{n-1} = \frac{1}{x} + \sum_{n=0}^{\infty} (n+1)! x^n\,. \tag{16.104}$$

$$\therefore\ (n+1)! = \left[x^n\right] \frac{F(x)-1}{x}\,. \tag{16.105}$$

Therefore,

$$\begin{aligned}
\left[x^n\right] H(x) &= n \cdot n! && (16.106)\\
&= (n+1-1) \cdot n! && (16.107)\\
&= (n+1)! - n! && (16.108)\\
&= \left[x^n\right] \frac{F(x)-1}{x} - \left[x^n\right] F(x) && (16.109)\\
&= \left[x^n\right] \left(\frac{F(x)-1}{x} - F(x) \right). && (16.110)
\end{aligned}$$

In other words,

$$H(x) = \frac{F(x)-1}{x} - F(x)\,. \tag{16.111}$$

$$\therefore\ F(x) = \frac{xH(x)+1}{1-x}\,. \tag{16.112}$$

$$xH(x)+1 = \sum_{n=0}^{\infty} n \cdot n! x^{n+1} + 1 = 1 + \sum_{n=1}^{\infty} (n-1) \cdot (n-1)! x^n\,. \tag{16.113}$$

$[x^n]\,(xH(x)+1)G(x)$ is 1 for $n = 0$ and is $(n-1) \cdot (n-1)!$ for $n \geq 1$. Let

$$G(x) = \frac{1}{1-x} = \sum_{n=0}^{\infty} x^n\,. \tag{16.114}$$

By the Convolution Rule,

$$\begin{aligned}
n! &= \left[x^n\right] F(x) && (16.115)\\
&= \left[x^n\right] (xH(x)+1)G(x) && (16.116)\\
&= 1 + \sum_{i=1}^{n} (i-1) \cdot (i-1)!\,. && (16.117)
\end{aligned}$$

□

IV PROBABILITY

17 Events and Probability Spaces

17.1 The Four Step Method

17.1.1 Terminologies

DEFINITION 17.1 **Outcome/Sample Point** ω An outcome consists all the information of an experiment after is has been performed, including the values of all random choices.

DEFINITION 17.2 **Sample Space** A set of all possible outcomes for an experiment.

DEFINITION 17.3 **Probability Function** Pr A total function Pr: $\mapsto \mathbb{R}$ such that
- $\forall\omega \in \ . 0 \leq \Pr(\omega) \leq 1$.
- $\sum_{\omega\in} \Pr(\omega) = 1$.

DEFINITION 17.4 **Probability Space** A sample space together with a probability function Pr.

DEFINITION 17.5 **Outcome Probability** $\Pr(\omega)$ The fraction of the time the outcome ω is expected to occur.

DEFINITION 17.6 **Event** E A subset of the sample space $E \subseteq \ $.

DEFINITION 17.7 **Discrete Event Probability** $\Pr\{E\}$ If is finite or countable infinite, then for any event E,

$$\Pr\{E\} = \sum_{\omega\in E} \Pr(\omega)\,. \tag{17.1}$$

DEFINITION 17.8 **Uniform Sample Space** A sample space is uniform if every outcome ω has the same outcome probability $\Pr(\omega) = \frac{1}{|\ |}$.

COROLLARY 17.9 If the sample space is uniform, for any event E

$$\Pr\{E\} = \frac{|E|}{|\ |}\,. \tag{17.2}$$

17.1.2 The Four Step Method

DEFINITION 17.10 **Tree Diagram** A tree to represent the probability space. The whole experiment is following the path from the root to a leaf, and the branch taken at each stage is randomly determined. The leaves of the tree represent outcomes of the experiment, and the set of all leaves represents the sample space. The edge weights denote the probability of that branch being taken given that we are at the parent of that branch.

The four step method is as follows

1. Find the sample space by using a tree diagram.
2. Define event of interest E by marking leaves corresponding to these outcomes.
3. Determine outcome probability $\Pr(\omega)$ by assigning edge probabilities and multiplying along root-to-leaf paths, where the edge probabilities are based on the assumptions we made.
4. Compute event probability $\Pr(E)$ by summing the probabilities of all outcomes in the event.

17.2 Monty Hall Problem

Suppose you are on a game show, and you are given the choice of three doors. Behind one door is a car, behind the others, goats. You pick a door, say door A, and the host, who knows what is behind the doors, opens another door, say door B, which has a goat. He says to you, "Do you want to pick door C?" Is it to your advantage to switch your choice of doors?

To avoid vagueness, we will assume that

- The car is equally likely to be hidden behind each of the three doors.
- The player is equally likely to pick each of the three doors, regardless of the car's location.
- After the player picks a door, the host must open a different door with a goat behind it and offer the player the choice of staying with the original door or switching.
- If the host has a choice of which door to open, then he is equally likely to select each of them.

THEOREM 17.11

$$\Pr\{\text{switching leads to win}\} = \frac{2}{3}. \tag{17.3}$$

In contrast,

$$\Pr\{\text{sticking leads to win}\} = \Pr\{\text{switching leads to lose}\} = \frac{1}{2}. \tag{17.4}$$

Proof. By using the four step method, see Fig. 17.1.

The outcome of an experiment of Monty Hall involves

1. The door concealing the car.
2. The door initially chosen by the player.
3. The door that the host opens to reveal a goat.

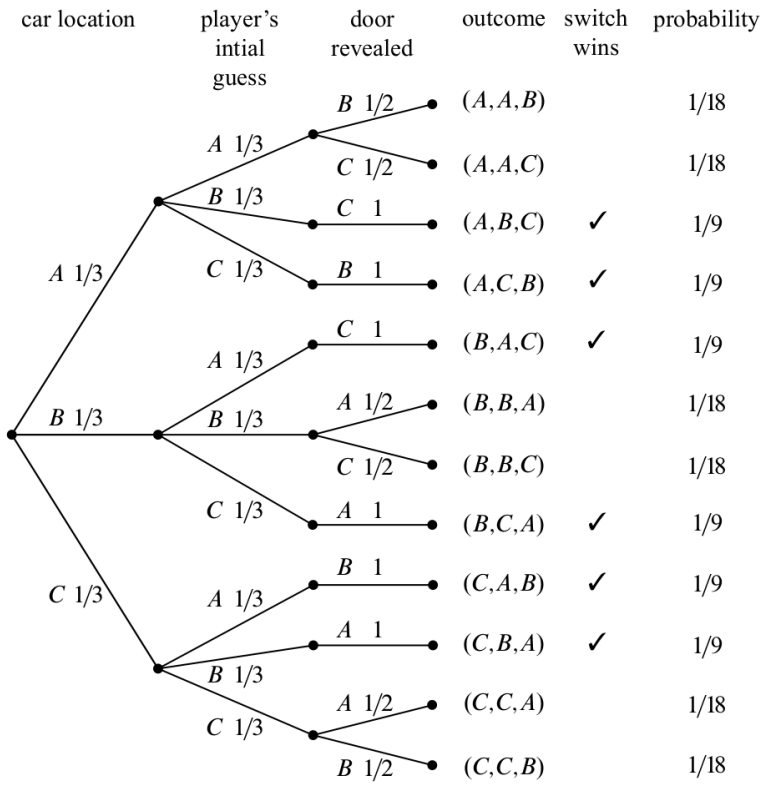

Figure 17.1
The tree diagram for the Monty Hall Problem.

We can represent each outcome by a triple of doors indicating

$$\text{(door concealing prize, door initially chosen, door opened to reveal a goat)}. \quad (17.5)$$

Define events of interest $E = \{\text{switching leads to win}\}$.
Therefore

$$\Pr\{E\} = \Pr((A,B,C)) + \Pr((A,C,B)) + \Pr((B,A,C)) + \quad (17.6)$$

$$\Pr((B,C,A)) + \Pr((C,A,B)) + \Pr((C,B,A)) \quad (17.7)$$

$$= 6 \times \frac{1}{9} \quad (17.8)$$

$$= \frac{2}{3}. \quad (17.9)$$

□

17.3 Strange Dice

17.3.1 Rolling Once

There are three strange dices and two players, each player selects one die and rolls it once. The player with the higher value wins. The three dices are as follows

- If you roll die A, $\Pr\{\text{get } 2\} = \Pr\{\text{get } 6\} = \Pr\{\text{get } 7\} = \frac{1}{3}$.
- If you roll die B, $\Pr\{\text{get } 1\} = \Pr\{\text{get } 5\} = \Pr\{\text{get } 9\} = \frac{1}{3}$.
- If you roll die C, $\Pr\{\text{get } 3\} = \Pr\{\text{get } 4\} = \Pr\{\text{get } 8\} = \frac{1}{3}$.

THEOREM 17.12

$$\Pr\{A \text{ beats } B\} = \frac{5}{9}. \tag{17.10}$$

Proof. By using the four-step method. Each outcome can be represents as a pair indicating

$$(\text{value rolled by die } A, \text{ value rolled by die } B). \tag{17.11}$$

Let the event

$$E = \{(i,j) \in \quad | \ i > j\}. \tag{17.12}$$

The tree diagram can be seen in Fig. 17.2.

Therefore,

$$\Pr\{E\} = 5 \times \frac{1}{9} = \frac{5}{9}. \tag{17.13}$$

□

By similar arguments, we can prove that

THEOREM 17.13

$$\Pr\{C \text{ beats } A\} = \frac{5}{9}, \tag{17.14}$$

$$\Pr\{B \text{ beats } C\} = \frac{5}{9}. \tag{17.15}$$

That is to say, "beats more often" is not a transitive binary relation.

17.3.2 Rolling Twice

This time, instead of rolling each die once, each player will roll die twice and sum the rolls as your score.

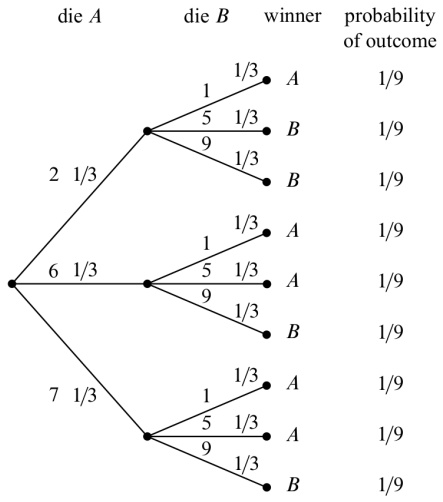

Figure 17.2
The tree diagram for one roll of die A versus die B.

THEOREM 17.14 If we roll die twice,

$$\Pr\{A \text{ beats } B\} = \frac{37}{81}, \tag{17.16}$$

$$\Pr\{B \text{ beats } A\} = \frac{37}{81}, \tag{17.17}$$

$$\Pr\{A \text{ ties } B\} = \frac{2}{81}, \tag{17.18}$$

$$\tag{17.19}$$

By using the four-step method. Each outcome can be represents as a pair indicating

$$(\text{sum of the two rolls of die } A, \text{ sum of the two rolls of die } B)\,. \tag{17.20}$$

Let the event

$$E := \{A \text{ beats } B\}\,. \tag{17.21}$$

If each player rolls twice, the tree diagram will have four levels and $3^4 = 81$ outcomes. This means that it will take a while to write down the entire tree diagram. But it is easy to write down the first two levels as in Fig. 17.3(a) and then notice that the remaining two levels consist of nine identical copies of the tree in Fig. 17.3(b).

Therefore

$$\Pr\{A \text{ beats } B\} = 42 \times \frac{1}{3^4} = \frac{42}{81}, \tag{17.22}$$

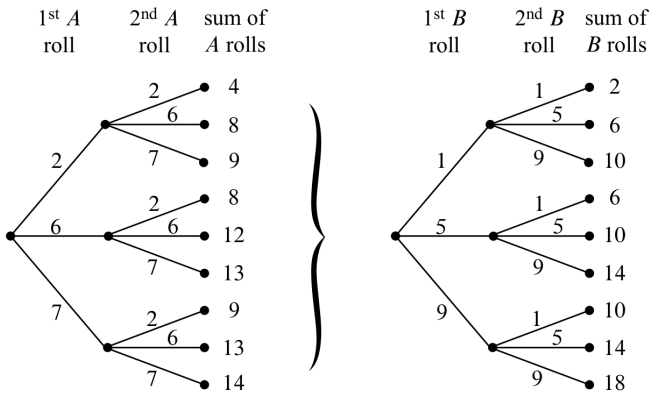

Figure 17.3

Parts of the tree diagram for die B versus die A where each die is rolled twice. The first two levels are shown in (a). The last two levels consist of nine copies of the tree in (b).

$$\Pr\{B \text{ beats } A\} = 37 \times \frac{1}{3^4} = \frac{37}{81}. \tag{17.23}$$

By similar arguments, we can prove that

THEOREM 17.15

$$\Pr\{C \text{ beats } A\} < \Pr\{A \text{ beats } C\}, \tag{17.24}$$

$$\Pr\{B \text{ beats } C\} < \Pr\{C \text{ beats } B\}. \tag{17.25}$$

Actually, there are arbitrarily large sets of dice which will beat each other in any desired pattern according to how many times the dice are rolled.

17.4 Set Theory and Probability

17.4.1 Probability Rules from Set Theory

THEOREM 17.16 **Sum Rule** If E_i's are pairwise disjoint events, then

$$\Pr\left\{\bigcup_i E_i\right\} = \sum_i \Pr\{E_i\}. \tag{17.26}$$

COROLLARY 17.17 **Complement Rule**

$$\Pr\{\bar{E}\} = 1 - \Pr\{E\}. \tag{17.27}$$

Proof. Because E and $\bar{E}$ are disjoint, by the Sum Rule,

$$\Pr\{E\} + \Pr\{\bar{E}\} = \Pr\{E \cup \bar{E}\} = \Pr\{\quad\} = 1. \tag{17.28}$$

□

COROLLARY 17.18 **Difference Rule**

$$\Pr\{B - A\} = \Pr\{B\} - \Pr\{A \cap B\}. \tag{17.29}$$

Proof. Because $B - A$ and $A \cap B$ are disjoint, by the Sum Rule,

$$\Pr\{B\} = \Pr\{(B - A) \cup (A \cap B)\} = \Pr\{B - A\} + \Pr\{A \cap B\}. \tag{17.30}$$

□

COROLLARY 17.19 **Inclusion-Exclusion**

$$\Pr\{A \cup B\} = \Pr\{A\} + \Pr\{B\} - \Pr\{A \cap B\}. \tag{17.31}$$

Proof. Because A and $B - A$ are disjoint, by the Sum Rule and Difference Rule,

$$\Pr\{A \cup B\} = \Pr\{A \cup (B - A)\} = \Pr\{A\} + \Pr\{B - A\} = \Pr\{A\} + \Pr\{B\} - \Pr\{A \cap B\}. \tag{17.32}$$

□

COROLLARY 17.20 **Boole's Inequality**

$$\Pr\{A \cup B\} \leq \Pr\{A\} + \Pr\{B\}. \tag{17.33}$$

Proof. By Inclusion-Exclusion

$$\Pr\{A \cup B\} = \Pr\{A\} + \Pr\{B\} - \Pr\{A \cap B\} \leq \Pr\{A\} + \Pr\{B\}, \tag{17.34}$$

since $\Pr\{A \cap B\} \geq 0$. □

COROLLARY 17.21 **Monotonicity Rule**

$$A \subseteq B \Rightarrow \Pr\{A\} \leq \Pr\{B\}. \tag{17.35}$$

Proof. It follows from the definition of event probability and the fact that outcome probabilities are nonnegative. □

COROLLARY 17.22 **Union Bound**

$$\Pr\left\{\bigcup_i E_i\right\} \leq \sum_i \Pr\{E_i\}. \tag{17.36}$$

17.5 The Birthday Principle

LEMMA 17.23

$$\forall x > 0.\ 1 - x < \exp(-x). \tag{17.37}$$

Proof. By using the Taylor series

$$\exp(-x) = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} > 1 - x. \tag{17.38}$$

□

THEOREM 17.24 **Birthday Principle** If there are n days in a year and $k = \sqrt{2n}$ people in a room, then the probability that two share a birthday is about $1 - \frac{1}{e} \approx 0.632$.

Proof. Assume that the probability that a randomly chosen people has a given birthday is $\frac{1}{n}$, and those k people is randomly and independently selected.

Each outcome ω can be represented as a length k sequence of their birthdays. Let the event E:={everyone has a different birthday}. There are $|\ \ | = n^k$ length k sequences of birthdays for k people, and under our assumptions, these are equally likely, which means that the sample space is uniform. There are $|E| = k!\binom{n}{k}$ length k sequences of distinct birthdays.

Therefore

$$\Pr\{E\} = \frac{|E|}{|\ \ |} \tag{17.39}$$

$$= \frac{k!\binom{n}{k}}{n^k} \tag{17.40}$$

$$= \frac{n(n-1)(n-2)\cdots(n-(k-1))}{n^k} \tag{17.41}$$

$$= \frac{n}{n}\frac{n-1}{n}\cdots\frac{n-(k-1)}{n} \tag{17.42}$$

$$= \left(1-\frac{0}{n}\right)\left(1-\frac{1}{n}\right)\cdots\left(1-\frac{k-1}{n}\right) \tag{17.43}$$

$$< \exp(-0)\exp(-\frac{1}{n})\cdots\exp(-\frac{k-1}{n}) \tag{17.44}$$

$$< \exp\left(-\sum_{i=1}^{k-1}\frac{i}{n}\right) \tag{17.45}$$

$$< \exp\left(-\frac{k(k-1)}{2n}\right). \tag{17.46}$$

When $k = \sqrt{2n}$,

$$\Pr\{E\} = \exp\left(-\frac{k(k-1)}{k^2}\right) \approx \frac{1}{e}. \tag{17.47}$$

□

It implies that to use a hash function that maps k items into a hash table of size n, you can expect many collisions if k^2 is more than a small fraction of n. The Birthday Principle also famously comes into play as the basis of birthday attacks that crack certain cryptographic systems.

17.6 Infinite Probability Spaces

Suppose two players take turns flipping a fair coin. Whoever flips heads first is declared the winner.

THEOREM 17.25

$$\Pr\{\text{the first player wins}\} = \frac{2}{3}. \tag{17.48}$$

$$\Pr\{\text{the second player wins}\} = \frac{2}{3}. \tag{17.49}$$

Proof. By the four-step method, see Fig. 17.4. The sample space

$$= \{T^n H \mid n \in \mathbb{N}\}, \tag{17.50}$$

where T^n stands for a length n string of T's. The probability function is

$$\Pr(T^n H) = \frac{1}{2^{n+1}}. \tag{17.51}$$

Therefore

$$\Pr\{\text{the first player wins}\} = \frac{1}{2} + \frac{1}{8} + \frac{1}{32} + \frac{1}{128} + \cdots \tag{17.52}$$

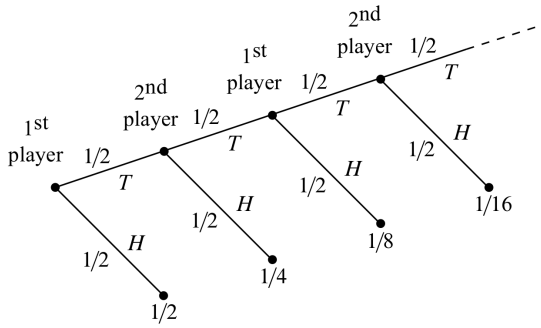

Figure 17.4
The tree diagram for the game where players take turns flipping a fair coin. The first player to flip heads wins.

$$= \frac{1}{2}\sum_{n=0}^{\infty}\left(\frac{1}{4}\right)^n \tag{17.53}$$

$$= \frac{1}{2}\left(\frac{1}{1-\frac{1}{4}}\right) \tag{17.54}$$

$$= \frac{2}{3}. \tag{17.55}$$

$$\Pr\{\text{the second player wins}\} = \frac{1}{4} + \frac{1}{16} + \frac{1}{64} + \cdots \tag{17.56}$$

$$= \frac{1}{4}\sum_{n=0}^{\infty}\left(\frac{1}{4}\right)^n \tag{17.57}$$

$$= \frac{1}{4}\left(\frac{1}{1-\frac{1}{4}}\right) \tag{17.58}$$

$$= \frac{1}{3}. \tag{17.59}$$

□

18 Conditional Probability

18.1 Definition and Notation

18.1.1 Definition

DEFINITION 18.1 **Conditional Probability** The probability of event A, given that event B happens is that

$$\Pr\{A \mid B\} := \frac{\Pr\{A \cap B\}}{\Pr\{B\}}. \tag{18.1}$$

The conditional probability $\Pr\{A \mid B\}$ is undefined when the probability of event B is zero.

When we know that B has occurred, every outcome that is outside B should be discarded. Thus, our sample space is reduced to the set B, see Fig. 18.1. Now the only way that A can happen is when the outcome belongs to the set $A \cap B$. We divide $\Pr\{A \cap B\}$ by $\Pr\{B\}$, so that the conditional probability of the new sample space becomes 1, i.e., $\Pr\{B \mid B\} = \frac{\Pr\{B \cap B\}}{\Pr\{B\}} = 1$.

18.1.2 Product Rule

COROLLARY 18.2 **Product Rule**

$$\Pr\{A \cap B\} = \Pr\{A\} \Pr\{B \mid A\} = \Pr\{B\} \Pr\{A \mid B\}. \tag{18.2}$$

Proof. It follows from the definition of conditional probability. □

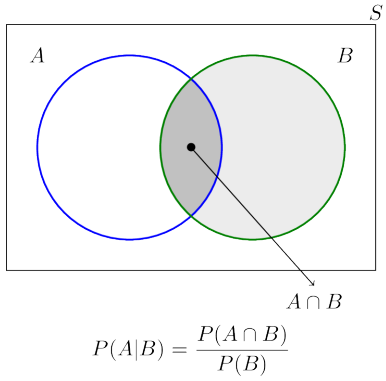

Figure 18.1
Conditional probability.

COROLLARY 18.3 **General Product Rule**

$$\Pr\left\{\bigcap_{i=1}^{n} A_i\right\} = \Pr\{A_1\}\prod_{i=2}^{n}\Pr\left\{A_i \mid \bigcap_{j=1}^{i-1} A_j\right\} \tag{18.3}$$

$$= \Pr\{A_1\}\Pr\{A_2 \mid A_1\}\Pr\{A_3 \mid A_1 \cap A_2\}\cdots\Pr\left\{A_n \mid \bigcap_{i=1}^{n-1} A_i\right\}. \tag{18.4}$$

Proof. By induction on n. The inductive hypothesis is

$$P(n) := \Pr\left\{\bigcap_{i=1}^{n} A_i\right\} = \Pr\{A_1\}\prod_{i=2}^{n}\Pr\left\{A_i \mid \bigcap_{j=1}^{i-1} A_j\right\}. \tag{18.5}$$

Base case: when $n = 1$,

$$\Pr\{A_1\} = \Pr\{A_1\}. \tag{18.6}$$

Inductive step: suppose $P(n)$ is true, therefore

$$\Pr\left\{\bigcap_{i=1}^{n+1} A_i\right\} = \Pr\left\{\left(\bigcap_{i=1}^{n} A_i\right) \cap A_{n+1}\right\} \tag{18.7}$$

$$= \Pr\left\{\bigcap_{i=1}^{n} A_i\right\}\Pr\left\{A_{n+1} \mid \bigcap_{j=1}^{n} A_i\right\} \tag{18.8}$$

$$= \Pr\{A_1\}\prod_{i=2}^{n}\Pr\left\{A_i \mid \bigcap_{j=1}^{i-1} A_j\right\}\Pr\left\{A_{n+1} \mid \bigcap_{j=1}^{n} A_i\right\} \tag{18.9}$$

$$= \Pr\{A_1\}\prod_{i=2}^{n+1}\Pr\left\{A_i \mid \bigcap_{j=1}^{i-1} A_j\right\}. \tag{18.10}$$

By the induction principle, $P(n)$ holds true for all n. □

18.1.3 The Law of Total Probability

The Law of Total Probability can let you reason about probabilities by cases.

THEOREM 18.4 **Law of Total Probability** For two event A and B, if $0 < \Pr\{B\} < 1$,

$$\Pr\{A\} = \Pr\{A \mid B\}\Pr\{B\} + \Pr\{A \mid \bar{B}\}\Pr\{\bar{B}\}. \tag{18.11}$$

Proof.

$$\Pr\{A \mid B\}\Pr\{B\} + \Pr\{A \mid \bar{B}\}\Pr\{\bar{B}\} = \Pr\{A \cap B\} + \Pr\{A \cap \bar{B}\} = \Pr\{A\}. \quad (18.12)$$

□

THEOREM 18.5 **General Law of Total Probability** For event A, and n disjoint events B_i, if $\forall i.\ 0 < \Pr\{B_i\} < 1$ and $\bigcup_{i=1}^{n} B_i =$,

$$\Pr\{A\} = \sum_{i=1}^{n} \Pr\{A \mid B_i\}\Pr\{B_i\}. \quad (18.13)$$

Proof.

$$\sum_{i=1}^{n} \Pr\{A \mid B_i\}\Pr\{B_i\} = \sum_{i=1}^{n} \Pr\{A \cap B_i\} = \Pr\{A\}. \quad (18.14)$$

□

18.1.4 Conditioning on a Single Event

The probability rules that we derived in the previous chapter extend to probabilities conditioned on the same event. For example, the Inclusion-Exclusion formula for two sets holds when all probabilities are conditioned on an event C:

THEOREM 18.6

$$\Pr\{A \cup B \mid C\} = \Pr\{A \mid C\} + \Pr\{B \mid C\} - \Pr\{A \cap B \mid C\}. \quad (18.15)$$

Proof.

$$\Pr\{A \cup B \mid C\} = \frac{\Pr\{(A \cup B) \cap C\}}{\Pr\{C\}} \quad (18.16)$$

$$= \frac{\Pr\{(A \cap C) \cup (B \cap C)\}}{\Pr\{C\}} \quad (18.17)$$

$$= \frac{\Pr\{A \cap C\} + \Pr\{B \cap C\} - \Pr\{A \cap B \cap C\}}{\Pr\{C\}} \quad (18.18)$$

$$= \Pr\{A \mid C\} + \Pr\{B \mid C\} - \Pr\{A \cap B \mid C\}. \quad (18.19)$$

□

It is important not to mix up events before and after the conditioning bar.

THEOREM 18.7

$$\Pr\{A \mid B \cup C\} \neq \Pr\{A \mid B\} + \Pr\{A \mid C\} - \Pr\{A \mid B \cap C\}. \quad (18.20)$$

Proof. For example, if $B.C \subseteq A$, and B, C are disjoint, then

$$\Pr\{A \mid B \cup C\} = \Pr\{A \mid B\} = \Pr\{A \mid C\} = 1\,. \tag{18.21}$$

$$\Pr\{A \mid B \cap C\} = 0\,. \tag{18.22}$$

Therefore,

$$1 \neq 1 + 1 - 0 = 2\,. \tag{18.23}$$

□

18.1.5 Probability of Size-k Subsets

THEOREM 18.8 Let's pick some size-k subset $S \subseteq [1,n]$ a target. Suppose we choose a size-k subset at random, with all subsets of $[1,n]$ equally likely to be chosen, then

$$\Pr\{\text{the randomly chosen subset equals the target } S\} = \frac{1}{\binom{n}{k}}\,. \tag{18.24}$$

In other words, the number of size-k subsets equals $\binom{n}{k}$.

Proof. Let

$$E_1 := \{\text{the smallest number in the random set is one of the } k \text{ numbers in } S\}\,, \tag{18.25}$$

$$E_2 := \{\text{the second smallest number in the random set is one of the } k-1 \text{ numbers in } S\}, \tag{18.26}$$

$$\ldots \tag{18.27}$$

$$E_k := \{\text{the largest number in the random set is in } S\}\,. \tag{18.28}$$

$$\Pr\{E_1\} = \frac{k}{n}\,, \tag{18.29}$$

$$\Pr\{E_2 \mid E_1\} = \frac{k-1}{n-1}\,, \tag{18.30}$$

$$\Pr\{E_3 \mid E_1 \cap E_2\} = \frac{k-2}{n-2}\,, \tag{18.31}$$

$$\ldots \tag{18.32}$$

$$\Pr\left\{E_k \mid \bigcap_{i=1}^{k-1} E_i\right\} = \frac{k-(k-1)}{n-(k-1)}\,, \tag{18.33}$$

$$\tag{18.34}$$

Therefore

$$\Pr\{\text{the randomly chosen subset equals the target } S\} \tag{18.35}$$

$$= \Pr\{E_1\} \prod_{i=2}^{k} \Pr\left\{ E_i \mid \bigcap_{j=1}^{i-1} E_j \right\} \tag{18.36}$$

$$= \frac{k}{n} \prod_{i=2}^{k} \frac{k-(i-1)}{n-(i-1)} \tag{18.37}$$

$$= \frac{k!(n-k)!}{n!} \tag{18.38}$$

$$= \frac{1}{\binom{n}{k}} . \tag{18.39}$$

□

18.2 A Posteriori Probabilities

18.2.1 Medical Testing

THEOREM 18.9 Suppose

$$\Pr\{y = 1\} = 1\% , \tag{18.40}$$

and

$$FNR = \Pr\{h(\vec{x}) = 0 \mid y = 1\} = 10\% , \tag{18.41}$$

$$FPR = \Pr\{h(\vec{x}) = 1 \mid y = 0\} = 5\% . \tag{18.42}$$

Then

$$\Pr\{y = 1 \mid h(\vec{x}) = 1\} = 15.4\% , \tag{18.43}$$

$$\Pr\{h(\vec{x}) \neq y\} = 95\% . \tag{18.44}$$

That is to say, if the test is positive, then there is an 84.6% chance that the result is incorrect, even though the test is nearly 95% accurate.

Proof. By using the four-step method. See Fig. 18.2. Then

$$\Pr\{y = 1 \mid h(\vec{x}) = 1\} = \frac{\Pr\{y = 1 \wedge h(\vec{x}) = 1\}}{\Pr\{h(\vec{x})\}} = \frac{0.0090}{0.0495 + 0.0090} = 15.4\% , \tag{18.45}$$

$$\Pr\{h(\vec{x}) \neq y\} = 0.9405 + 0.0090 = 95\% . \tag{18.46}$$

□

This is because the number of healthy individuals m_- is so large that even the mere 5% with false positive results overwhelm the number of genuinely positive results from the truly ill m_+.

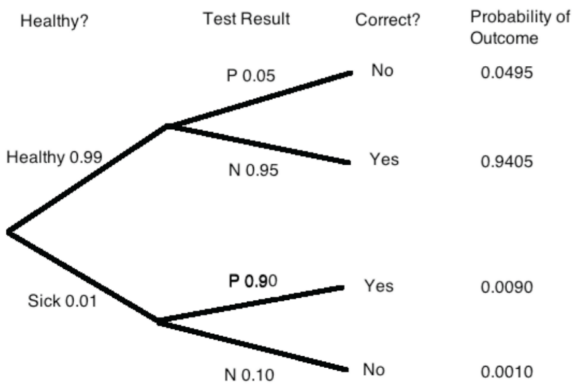

Figure 18.2
The tree diagram for the medical test.

18.2.2 Bayes' Rule

DEFINITION 18.10 **A Posteriori** A conditional probability $\Pr\{A \mid B\}$ is called a posteriori if event A precedes event B in time.

THEOREM 18.11 **Bayes' Rule**

$$\Pr\{A \mid B\} = \frac{\Pr\{B \mid A\}\Pr\{A\}}{\Pr\{B\}} = \frac{\Pr\{B \mid A\}\Pr\{A\}}{\sum_A \Pr\{B \mid A\}\Pr\{A\}}. \tag{18.47}$$

Proof. By the Product Rule,

$$\Pr\{A \cap B\} = \Pr\{A\}\Pr\{B \mid A\} = \Pr\{B\}\Pr\{A \mid B\}. \tag{18.48}$$

□

18.3 Simpson's Paradox

DEFINITION 18.12 **Simpson's Paradox** Case where multiple small groups of data all exhibit a similar trend, but that trend reverses when those groups are aggregated.

For example, consider the experiment where we pick a random candidate, and assume that all candidates are either men or women, and that no candidate belongs to both departments. define the following events,

$$A := \{\text{the candidate is admitted to his/her program of choice}\}, \tag{18.49}$$

$$F_{EN} := \{\text{the candidate is a woman applying to the English department}\}, \quad (18.50)$$

$$F_{CS} := \{\text{the candidate is a woman applying to the CS department}\}, \quad (18.51)$$

$$M_{EN} := \{\text{the candidate is a man applying to the English department}\}, \quad (18.52)$$

$$M_{CS} := \{\text{the candidate is a man applying to the CS department}\}. \quad (18.53)$$

Consider the case

$$\Pr\{A \mid M_{EN}\} = \frac{2}{5} = 40\%, \quad (18.54)$$

$$\Pr\{A \mid F_{EN}\} = \frac{50}{100} = 50\%, \quad (18.55)$$

$$\Pr\{A \mid M_{CS}\} = \frac{70}{100} = 70\%, \quad (18.56)$$

$$\Pr\{A \mid F_{CS}\} = \frac{4}{5} = 80\%. \quad (18.57)$$

Therefore for any given department, a male applicant is less likely to be admitted than a female:

$$\Pr\{A \mid M_{EN}\} < \Pr\{A \mid F_{EN}\}, \quad (18.58)$$

$$\Pr\{A \mid M_{CS}\} < \Pr\{A \mid F_{CS}\}, \quad (18.59)$$

while as a total count, a male candidate is more likely to be admitted to the university than a female:

$$\Pr\{A \mid M_{EN} \cup M_{CS}\} = \frac{72}{105} = 69\% > \Pr\{A \mid F_{EN} \cup F_{CS}\} = \frac{54}{105} = 51\%. \quad (18.60)$$

The reason is that, the CS department is more selective for man than the English department, but English attracts a far larger number of woman applicants than CS.

We interpreted the same numbers differently based on our implicit causal beliefs, it is circular to claim that the data corroborated our beliefs that there is or is not sex discrimination. Rather, our interpretation of the data correlation depended on our beliefs about the causes of admission in the first place. This example highlights a basic principle in statistics which people constantly ignore: *never assume that correlation implies causation*.

18.4 Independence

18.4.1 Independence

DEFINITION 18.13 **Independence** An event with probability 0 is defined to be independent of every event (including itself). If $\Pr\{B\} \neq 0$, then event A is independent of event B iff

$$\Pr\{A \mid B\} = \Pr\{A\}. \quad (18.61)$$

In other words, A and B are independent if knowing that B happens does not alter the probability that A happens.

Generally, independence is something that you assume in modeling a phenomenon.

LEMMA 18.14 Disjoint events are never independent given that no event is empty.

Proof. If events A and B are disjoint, i.e., $A \cap B = \varnothing$. Then

$$\Pr\{A \mid B\} = \frac{\Pr\{A \cap B\}}{\Pr\{B\}} = 0 \neq \Pr\{B\}. \tag{18.62}$$

□

THEOREM 18.15 A is independent of B iff

$$\Pr\{A \cap B\} = \Pr\{A\}\Pr\{B\}. \tag{18.63}$$

Proof.

$$\Pr\{A \cap B\} = \Pr\{A \mid B\}\Pr\{B\} = \Pr\{A\}\Pr\{B\} \tag{18.64}$$

iff A is independent of B. □

COROLLARY 18.16 A is independent of B iff B is independent of A.

18.4.2 Mutual Independence

DEFINITION 18.17 **Mutual Independence** Events $A_1, A_2, \ldots, A_n$ are mutually independent iff the probability of each event in the set is the same no matter which of the other events has occurred. To be more precise,

$$\forall i \in [1,n], \forall S \subseteq ([1,n] - \{i\}).\ \Pr\left\{A_i \mid \bigcap_{j \in S} A_j\right\} = \Pr\{A_i\}. \tag{18.65}$$

THEOREM 18.18 Events $A_1, A_2, \ldots, A_n$ are mutually independent iff

$$\forall S \subseteq [1,n].\ \Pr\left\{\bigcap_{i \in S} A_i\right\} = \prod_{i \in S} \Pr\{A_i\}. \tag{18.66}$$

18.4.3 Pairwise Independence

DEFINITION 18.19 ***k*-way Independent** Events $A_1, A_2, \ldots, A_n$ are k-way independent iff every set of k of these events is mutually independent.

DEFINITION 18.20 **Pairwise Independent** The set is pairwise independent iff it is 2-way independent.

THEOREM 18.21 For example, suppose that we flip three fair, mutually-independent coins. Define the following events

$$A_1 := \{\text{result of coin 1 matches coin 2}\}, \tag{18.67}$$

$$A_2 := \{\text{result of coin 2 matches coin 3}\}, \tag{18.68}$$

$$A_3 := \{\text{result of coin 3 matches coin 1}\}. \tag{18.69}$$

Then A_1, A_2, A_3 are pairwise independent but not mutually independent.

Proof. The sample space is

$$= \{(\text{HHH}), (\text{HHT}), (\text{HTH}), (\text{HTT}), (\text{THH}), (\text{THT}), (\text{TTH}), (\text{TTT})\}. \tag{18.70}$$

$$\Pr\{A_1\} = \Pr((\text{HHH})) + \Pr((\text{HHT})) + \Pr((\text{TTH})) + \Pr((\text{TTT})) = \frac{4}{8} = \frac{1}{2}. \tag{18.71}$$

By symmetry, $\Pr\{A_2\} = \Pr\{A_3\} = \frac{1}{2}$ as well.

$$\Pr\{A_1 \cap A_2\} = \Pr((\text{HHH})) + \Pr((\text{TTT})) = \frac{2}{8} = \frac{1}{4} = \Pr\{A_1\}\Pr\{A_2\}. \tag{18.72}$$

By symmetry,

$$\Pr\{A_1 \cap A_3\} = \Pr\{A_1\}\Pr\{A_3\}, \tag{18.73}$$

$$\Pr\{A_2 \cap A_3\} = \Pr\{A_2\}\Pr\{A_3\}. \tag{18.74}$$

$$\Pr\{A_1 \cap A_2 \cap A_3\} = \Pr((\text{HHH})) + \Pr((\text{TTT})) = \frac{2}{8} = \frac{1}{4} \neq \Pr\{A_1\}\Pr\{A_2\}\Pr\{A_3\}. \tag{18.75}$$

□

18.5 Philosophy of Probability

18.5.1 Frequentist

DEFINITION 18.22 **Frequentist** Probabilities can only be meaningfully applied to repeatable processes like rolling dice or flipping coins. The probability of an event represents the fraction of trials in which the event occurred.

DEFINITION 18.23 **Confidence** Confidence is usually used to describe the probability that a statistical estimations of some quantity is correct.

18.5.2 Bayesian

DEFINITION 18.24 **Bayesian** A probability is interpreted as a degree of belief in a proposition. Bayesian is willing to assign probabilities to any event, but the problem is that there is no single "right" probability for an event, since the probability depends on one's initial beliefs. On the other hand, if you have confidence in some set of initial beliefs, then Bayesianism provides a convincing framework for updating your beliefs as further information emerges.

DEFINITION 18.25 **Odds** For an event A, the odds of A is defined as

$$\text{odds}(A) := \frac{\Pr\{A\}}{\Pr\{\bar{A}\}} = \frac{\Pr\{A\}}{1 - \Pr\{A\}}. \tag{18.76}$$

COROLLARY 18.26

$$\Pr\{A\} = \frac{\text{odds}(A)}{1 + \text{odds}(A)}. \tag{18.77}$$

For example, if $\text{odds}(A) = \frac{1}{5}$, we say that the odds of A is "five to one". Besides, $\Pr\{A\} = \frac{1}{6}$.

DEFINITION 18.27 **Bayes Factor** $K(E \mid A)$ For events A and E, the Bayes factor is

$$K(E \mid A) := \frac{\Pr\{E \mid A\}}{\Pr\{E \mid \bar{A}\}}. \tag{18.78}$$

THEOREM 18.28 Suppose an event E offers some evidence about A. Then the odds of A given E is

$$\text{odds}(A \mid E) = K(E \mid A)\,\text{odds}(A). \tag{18.79}$$

Proof.

$$\text{odds}(A \mid E) = \frac{\Pr\{A \mid E\}}{\Pr\{\bar{A} \mid E\}} \tag{18.80}$$

$$= \frac{\frac{\Pr\{E|A\}\Pr\{A\}}{\Pr\{E\}}}{\frac{\Pr\{E|\bar{A}\}\Pr\{\bar{A}\}}{\Pr\{E\}}} \tag{18.81}$$

$$= \frac{\Pr\{E \mid A\}}{\Pr\{E \mid \bar{A}\}} \frac{\Pr\{A\}}{\Pr\{\bar{A}\}} \tag{18.82}$$

$$= K(E \mid A) \operatorname{odds}(A) . \tag{18.83}$$

□

19 Random Variables

19.1 Random Variables and Independence

19.1.1 Random Variables

DEFINITION 19.1 **Random Variable** A random variable X on a probability space is a total function $X\colon \quad \mapsto \mathbb{R}$.

A random variable partitions the sample space, each block corresponds to an event that $X = x$.

$$E_x = \{\omega \in \quad \mid X(\omega) = x\}\,. \tag{19.1}$$

THEOREM 19.2

$$\Pr\{X = x\} = \sum_{\omega : X(\omega) = x} \Pr(\omega)\,. \tag{19.2}$$

THEOREM 19.3 For a set $S \subseteq \mathbb{R}$,

$$\Pr\{X \in S\} = \sum_{x \in S} \Pr\{X = x\}\,. \tag{19.3}$$

DEFINITION 19.4 **Indicator Random Variable/Bernoulli Variable** A random variable that maps every outcome to either 0 or 1.

An event E partitions the sample space into those outcomes in E and those not in E. So E is naturally associated with an indicator random variable, I_E, where $I_E = 1$ for outcomes $\omega \in E$ and $I_E = 0$ for outcomes $\omega \notin E$.

19.1.2 Independence

DEFINITION 19.5 **Independence** Random variables X and Y are independent iff

$$\forall x, y \in \mathbb{R}.\ \Pr\{X = x \mid Y = y\} = \Pr\{X = x\}\,. \tag{19.4}$$

The independence of two random variables means that knowing some information about one variable does not provide any information about the other one.

THEOREM 19.6 Random variables X and Y are independent iff

$$\forall x, y \in \mathbb{R}.\ \Pr\{X = x \wedge Y = y\} = \Pr\{X = x\} \Pr\{Y = y\}\,. \tag{19.5}$$

LEMMA 19.7 Two events are independent iff their indicator variables are independent.

LEMMA 19.8 Let X and Y be independent random variables, and f and g be functions such that $\operatorname{dom} f = \operatorname{cod} X$ and $\operatorname{dom} g = \operatorname{cod} Y$. Then $f(X)$ and $g(Y)$ are independent random variables. In other words, functions of independent variables are also independent.

DEFINITION 19.9 **Mutual Independence** Random variables $X_1, X_2, \ldots, X_n$ are mutually independent iff

$$\forall x_1, x_2, \ldots, x_n \in \mathbb{R}.\ \Pr\{X_1 = x_1 \wedge X_2 = x_2 \wedge \cdots \wedge X_n = x_n\} = \prod_{i=1}^{n} \Pr\{X_i = x_i\}\,. \tag{19.6}$$

19.2 Distribution Functions

19.2.1 PMF and CDF

DEFINITION 19.10 **Probability Mass Function (PMF)** P Let X be a random variable with codomain R. The probability mass function of X is a function $P\colon \mathbb{R} \mapsto [0,1]$ defined by

$$P(x) := \begin{cases} \Pr\{X = x\} & \text{if } x \in \operatorname{ran} X; \\ 0 & \text{if } x \notin \operatorname{ran} X. \end{cases} \tag{19.7}$$

DEFINITION 19.11 **Cumulative Distribution Function (CDF)** F Let X be a random variable with codomain R. The cumulative distribution function of X is a function $F\colon \mathbb{R} \mapsto [0,1]$ defined by

$$F(x) := \Pr\{X \leq x\} = \sum_{i \leq x} P(i)\,. \tag{19.8}$$

LEMMA 19.12

$$\sum_{x} P(x) = 1\,. \tag{19.9}$$

Proof.

$$\sum_{x} P(x) = \sum_{x \in \operatorname{ran} X} \Pr\{X = x\} = \sum_{\omega \in} \Pr(\omega) = 1\,. \tag{19.10}$$

□

LEMMA 19.13

$$\lim_{x \to -\infty} F(x) = 0\,, \tag{19.11}$$

$$\lim_{x \to \infty} F(x) = 1\,. \tag{19.12}$$

19.2.2 Bernoulli Distribution

DEFINITION 19.14 **Bernoulli Trial** An experiment with only two possible outcomes: success, which occurs with probability p, and failure, which occurs with probability $1 - p$.

DEFINITION 19.15 **Bernoulli Distribution** A Bernoulli distribution is the distribution function for a Bernoulli variable. Specifically, the Bernoulli distribution has a pmf

$$P(x) := \begin{cases} p & \text{if } x = 0; \\ 1 - p & \text{if } x = 1; \\ 0 & \text{otherwise,} \end{cases} \tag{19.13}$$

for some $p \in [0, 1]$. The corresponding cdf is

$$F(x) := \begin{cases} 0 & \text{if } x \in (-\infty, 0); \\ p & \text{if } x \in [0, 1); \\ 1 & \text{if } x \in [1, \infty). \end{cases} \tag{19.14}$$

19.2.3 Uniform Distribution

19.2.3.1 Uniform Distribution

DEFINITION 19.16 **Uniform** A random variable that takes on each possible value in its codomain with the same probability.

DEFINITION 19.17 **Uniform Distribution** If the codomain of a random variable has n numbers in increasing order being $a_1, a_2, \ldots, a_n$, then the pmf is

$$P(x) := \begin{cases} \frac{1}{n} & \text{if } x \in \{a_1, a_2, \ldots, a_n\}; \\ 0 & \text{otherwise.} \end{cases} \tag{19.15}$$

The corresponding cdf is

$$F(x) := \begin{cases} 0 & \text{if } x \in (-\infty, a_1); \\ \frac{k}{n} & \text{if } x \in \left[a_k, a_{k+1}\right) \text{ for } 1 \leq k < n; \\ 1 & \text{if } x \in [a_n, \infty). \end{cases} \tag{19.16}$$

19.2.3.2 Randomized Algorithms

DEFINITION 19.18 **Randomized Algorithm** Algorithm involve random numbers to influence decisions.

We have two envelopes. Each contains an integer in the range $[0, n]$, and the numbers are distinct. To win the game, you must determine which envelope contains the larger number. To give you a fighting chance, we will let you peek at the number in one envelope selected at random.

The numbers in the envelopes may not be random. Wlog, we denote the lower number a and higher number b. We are picking the numbers and we are choosing them in a way that we think will defeat your guessing strategy, i.e., to let you lose.

Suppose that you somehow knew a number x that was in between the numbers in the envelopes. Now you peek in one envelope and see a number. If it is bigger than x, then you know you are peeking at the higher number. If it is smaller than x, then you are peeking at the lower number. That means, you are certain to win the game.

The only flaw with this brilliant strategy is that you do not know such an x, but there is a way to salvage things: try to guess x. That is to say, your goal is to guess a number x between a and b, so you should select x at random from among the half-integers in uniform:

$$\frac{1}{2}, 1 + \frac{1}{2}, \ldots, (n-1) + \frac{1}{2}. \tag{19.17}$$

The reason you select uniformly is that if we figured out that you were unlikely to pick some number, say $50 + \frac{1}{2}$, then we would always put 50 and 51 in the envelopes. Then you would be unlikely to pick an x between a and b and would have less chance of winning.

There is some probability that you guess correctly. In this case, you win 100% of the time. On the other hand, if you guess incorrectly, then you are no worse off than before; your chance of winning is still 50%. Combining these two cases, your overall chance of winning is better than 50%.

THEOREM 19.19 The probability you win by using the guessing x strategy is greater than 50%.

Proof. By using the four-step method. See Fig. 19.1.

$$\Pr\{\text{win}\} = \frac{a}{2n} + \frac{b-a}{2n} + \frac{b-a}{2n} + \frac{n-b}{2n} = \frac{1}{2} + \frac{b-a}{2n} \geq \frac{1}{2} + \frac{1}{2n}. \tag{19.18}$$

□

19.2.4 Geometric Distribution

Suppose we have a sequence of independent Bernoulli trials, each with a probability p of success and probability $1 - p$ of failure. Let X be the number of trials need to obtain a success, then $X \sim Geo(p)$.

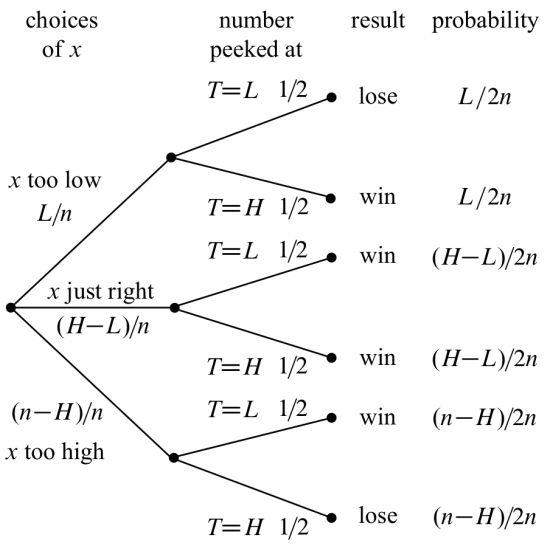

Figure 19.1
The tree diagram for the numbers game.

DEFINITION 19.20 **Geometric Distribution** $Geo(p)$ For some $p \in [0,1]$, the Geometric distribution has a pmf

$$P(x) := \begin{cases} (1-p)^{x-1}p & \text{if } x \in \mathbb{N}^+; \\ 0 & \text{otherwise.} \end{cases} \tag{19.19}$$

The corresponding cdf is

$$F(x) := \begin{cases} 0 & \text{if } x \in (-\infty, 1); \\ \sum_{i=1}^{k}(1-p)^{i-1}p & \text{if } x \in [k, k+1) \text{ for } 1 \leq k < n. \end{cases} \tag{19.20}$$

19.2.5 Binomial Distribution

19.2.5.1 Binomial Distribution

Suppose we have n independent Bernoulli trials, each with a probability p of success and probability $1-p$ of failure. Let X be the number of succeed trials, then $X \sim Bin(n,p)$.

DEFINITION 19.21 **Binomial Distribution** $Bin(n,p)$ For some $n \in \mathbb{N}^+$ and $p \in [0,1]$, the Binomial distribution has a pmf

$$P(x) := \begin{cases} \binom{n}{x}p^x(1-p)^{n-x} & \text{if } x \in [0,n]; \\ 0 & \text{otherwise.} \end{cases} \tag{19.21}$$

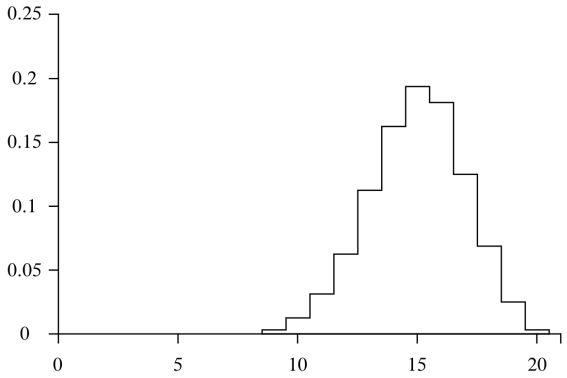

Figure 19.2
The pmf for the binomial distribution with $n = 20, p = 0.75$.

The corresponding cdf is

$$F(x) := \begin{cases} 0 & \text{if } x \in (-\infty, 0); \\ \sum_{i=0}^{k} \binom{n}{i} p^i (1-p)^{n-i} & \text{if } x \in [k, k+1) \text{ for } 0 \le k < n; \\ 1 & \text{if } x \in [n, \infty). \end{cases} \tag{19.22}$$

A plot of $p(x)$ with $n = 20, p = 0.75$ is shown in Fig. 19.2, The most likely outcome is $x = np$, and the probability falls off rapidly for larger and smaller values of x. The falloff regions to the left and right of the main hump are called the **tails of the distribution**.

19.2.5.2 Approximating the PMF and CDF

LEMMA 19.22 [Approximation of Binomial Coefficient]

$$\binom{n}{\alpha n} \sim \frac{2^{nH(\alpha)}}{\sqrt{2\pi\alpha(1-\alpha)n}}, \tag{19.23}$$

$$\binom{n}{\alpha n} < \frac{2^{nH(\alpha)}}{\sqrt{2\pi\alpha(1-\alpha)n}}, \tag{19.24}$$

where $H(\alpha)$ is the entropy function

$$H(\alpha) := \alpha \lg \frac{1}{\alpha} + (1-\alpha) \lg \frac{1}{1-\alpha}. \tag{19.25}$$

LEMMA 19.23 The pmf of Binomial distribution

$$P(x) \sim \frac{2^{nH(\alpha) p^{\alpha n} (1-p)^{(1-\alpha n)}}}{\sqrt{2\pi\alpha(1-\alpha)n}} = \frac{2^{n(\alpha \lg \frac{p}{\alpha} + (1-\alpha) \lg \frac{1-p}{1-\alpha})}}{\sqrt{2\pi\alpha(1-\alpha)n}}, \tag{19.26}$$

$$P(x) < \frac{2^{n(\alpha \lg \frac{p}{\alpha} + (1-\alpha)\lg \frac{1-p}{1-\alpha})}}{\sqrt{2\pi\alpha(1-\alpha)n}}. \tag{19.27}$$

COROLLARY 19.24

$$P(pn) \sim \frac{1}{\sqrt{2\pi p(1-p)n}}. \tag{19.28}$$

LEMMA 19.25 For $\alpha < p$,

$$F(\alpha n) < \frac{1-\alpha}{1-\frac{\alpha}{p}} P(\alpha n). \tag{19.29}$$

If $\alpha < p$ is not the case in your problem, then try thinking in complementary terms; that is, look at the number of tails flipped instead of the number of heads.

19.2.5.3 Noisy Channels

Suppose you are sending $n = 10^4$ packets of data across a communication channel and that each packet is lost with probability $p = 0.01$. Also suppose that packet losses are independent. You need to figure out how much redundancy (or error correction) to build into your communication protocol. Would it be safe for you to assume that only 2% are lost?

$$\Pr\{\text{number lost} \geq 0.02n\} = \Pr\{\text{number not lost} \leq 0.98n\} \tag{19.30}$$

$$\leq \frac{0.98}{1-\frac{0.98}{0.99}} \frac{2^{10^4(0.98\lg\frac{0.99}{0.98} + 0.02\lg\frac{0.01}{0.02})}}{\sqrt{2\pi \times 0.98 \times (1-0.98) \times 10^4}} < 2^{-60} \tag{19.31}$$

It says that planning on at most 2% packet loss in a batch of 10^4 packets should be very safe.

19.2.5.4 Estimation by Sampling

DEFINITION 19.26 **Sampling** Used for estimating the fraction of elements in a set that have a certain property.

Suppose The fraction of Americans that intend to vote Republican is a fixed and unknown value p (that is not a random variable). We contact n Americans selected at random and then compute the fraction of those Americans that will vote Republican. This value is then used as the estimate of the number of all Americans that will vote Republican.

We assume that the i-th contacted American is selected uniformly at random (with replacement) from the set of all Americans. Let X_i be the indicator random variable where

$X_i = 1$ if the i-th contacted American intends to vote Republican and $X_i = 0$ otherwise. Thus

$$\Pr\{X_i = 1\} = p\,. \tag{19.32}$$

Let

$$X := \sum_{i=1}^{n} X_i \tag{19.33}$$

to be the number of contacted Americans who intend to vote Republican.

$$X \sim \mathrm{Bin}(n,p)\,. \tag{19.34}$$

Then $\frac{X}{n}$ is a random variable that is the estimate of the fraction of Americans that intend to vote Republican, which is called the **statistical estimation** of p.

The statement "There is a $1-\delta$ probability that the poll is accurate to within γ." means that

$$\Pr\left\{\left|\frac{X}{n} - p\right| \le \gamma\right\} \ge 1-\delta\,. \tag{19.35}$$

How many people n do we need to contact to make sure that above equation is true? Define

$$Y := n - X\,. \tag{19.36}$$

$$Y \sim \mathrm{Bin}(n, 1-p)\,. \tag{19.37}$$

Then

$$\begin{aligned}
\Pr\left\{\left|\frac{X}{n} - p\right| > \gamma\right\} &= \Pr\{X < (p-\gamma)n\} + \Pr\{X > (p+\gamma)n\} && (19.38)\\
&= \Pr\{X < (p-\gamma)n\} + \Pr\{Y < (1-p-\gamma)n\} && (19.39)\\
&= F_X((p-\gamma)n) + F_Y((1-p-\gamma)n) && (19.40)\\
&\le F_X((0.5-\gamma)n) + F_Y((0.5-\gamma)n) && (19.41)\\
&= 2F((0.5-\gamma)n) && (19.42)\\
&< 2\frac{1-(0.5-\gamma)}{1-\frac{0.5-\gamma}{0.5}} P((0.5-\gamma)n) && (19.43)\\
&< 2\frac{1-(0.5-\gamma)}{1-\frac{0.5-\gamma}{0.5}} \frac{2^{n(0.5-\gamma)\lg\frac{0.5}{0.5-\gamma} + (0.5+\gamma)\lg\frac{0.5}{0.5+\gamma}}}{\sqrt{2\pi(0.5-\gamma)(0.5+\gamma)n}} && (19.44)\\
&< \delta\,. && (19.45)
\end{aligned}$$

$\Pr\left\{\left|\frac{X}{n} - p\right| > \gamma\right\}$ is maximized when $p = 0.5$. Substituting γ and δ we can find n.

19.3 Expectations

19.3.1 Expectations

DEFINITION 19.27 **Expectation/Mean/Average μ** If X is a random variable defined on a sample space , then the expectation of X is

$$\mu := \mathbb{E}[X] := \sum_{\omega \in} X(\omega) \Pr(\omega) . \tag{19.46}$$

THEOREM 19.28 For any random variable X,

$$\mu = \mathbb{E}[X] = \sum_{x \in \operatorname{ran} X} x \cdot \Pr\{X = x\} . \tag{19.47}$$

Proof.

$$\begin{aligned}
\mathbb{E}[X] &= \sum_{\omega \in} X(\omega) \Pr(\omega) && (19.48)\\
&= \sum_{x \in \operatorname{ran} X} \sum_{\omega \in \{X = x\}} X(\omega) \Pr\{\omega\} && (19.49)\\
&= \sum_{x \in \operatorname{ran} X} \sum_{\omega \in \{X = x\}} x \Pr\{\omega\} && (19.50)\\
&= \sum_{x \in \operatorname{ran} X} x \left(\sum_{\omega \in \{X = x\}} \Pr\{\omega\} \right) && (19.51)\\
&= \sum_{x \in \operatorname{ran} X} x \Pr\{X = x\} . && (19.52)
\end{aligned}$$

□

COROLLARY 19.29 If $\operatorname{ran} X = \mathbb{N}$, then

$$\mathbb{E}[X] = \sum_{i=1}^{\infty} i \cdot \Pr\{X = x\} = \sum_{i=0}^{\infty} \Pr\{X > i\} . \tag{19.53}$$

Proof. The first equality follows directly from the Theorem and the fact that $\operatorname{ran} X = \mathbb{N}$. The second equality is derived by adding the following equations:

$$\begin{aligned}
\Pr\{X > 0\} &= \Pr\{X = 1\} + \Pr\{X = 2\} + \Pr\{X = 3\} + \cdots , && (19.54)\\
\Pr\{X > 1\} &= \Pr\{X = 2\} + \Pr\{X = 3\} + \cdots , && (19.55)\\
\Pr\{X > 2\} &= \Pr\{X = 3\} + \cdots , && (19.56)
\end{aligned}$$

$$\cdots \tag{19.57}$$

which gives

$$\sum_{i=0}^{\infty} \Pr\{X > i\} = 1 \cdot \Pr\{X = 1\} + 2 \cdot \Pr\{X = 2\} + 3 \cdot \Pr\{X = 3\} + \cdots \tag{19.58}$$

$$= \sum_{i=1}^{\infty} i \cdot \Pr\{X = x\} . \tag{19.59}$$

□

COROLLARY 19.30 For any random variable $X \in \mathbb{N}$,

$$\Pr\{X \geq 1\} \leq \mathbb{E}[X] . \tag{19.60}$$

Proof.

$$\mathbb{E}[X] = \sum_{i=0}^{\infty} \Pr\{X > i\} \geq \Pr\{X \geq 1\} . \tag{19.61}$$

□

THEOREM 19.31 When two random variables X and Y are independent and each has a defined expectation,

$$\mathbb{E}[XY] = \mathbb{E}[X]\,\mathbb{E}[Y] . \tag{19.62}$$

Proof.

$$\mathbb{E}[XY] = \sum_{x} \sum_{y} xy \Pr\{X = x \wedge Y = y\} \tag{19.63}$$

$$= \sum_{x} \sum_{y} xy \Pr\{X = x\} \Pr\{Y = y\} \tag{19.64}$$

$$= \left(\sum_{x} x \Pr\{X = x\} \right) \left(\sum_{y} y \Pr\{Y = y\} \right) \tag{19.65}$$

$$= \mathbb{E}[X]\,\mathbb{E}[Y] . \tag{19.66}$$

□

THEOREM 19.32 When n random variables $X_1, X_2, \ldots, X_n$ are mutually independent,

$$\mathbb{E}[X_1 X_2 \ldots X_n] = \prod_{i=1}^{n} \mathbb{E}[X_i] . \tag{19.67}$$

DEFINITION 19.33 **Convex Function** A function $f(x)$ is convex iff

$$\forall x, y \in \mathbb{R}, \forall 0 \leq \alpha \leq 1. f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha) f(y). \tag{19.68}$$

THEOREM 19.34 **Jensen's Inequality** For a convex function $f(x)$ and a random variable X,

$$\mathbb{E}[f(X)] \geq f(\mathbb{E}[X]), \tag{19.69}$$

provided that the expectations exist and are finite.

DEFINITION 19.35 **Median** The median of a random variable X is the value $x \in \operatorname{ran} X$ such that

$$\Pr\{X < x\} \leq \frac{1}{2} \wedge \Pr\{X > x\} < \frac{1}{2}. \tag{19.70}$$

19.3.2 Pitfall: Computing Expectations by Sampling

Suppose that you are trying to estimate a parameter such as the average delay across a communication channel. So you set up an experiment to measure how long it takes to send a test packet from one end to the other and you run the experiment 100 times, and then take the average

$$\bar{x} := \frac{1}{100} \sum_{i=1}^{100} x^{(i)}. \tag{19.71}$$

Suppose that $\bar{x} = 8.3$ ms.

In fact, the expected latency might well be infinite. Let X be a random variable that denotes the time it takes for the packet to cross the channel. Suppose that the pmf of X is

$$P(x) := \begin{cases} 0 & \text{if } x = 0; \\ \frac{1}{x} - \frac{1}{x+1} & \text{if } x \in \mathbb{N}^+. \end{cases} \tag{19.72}$$

We might expect that X is likely to be small. It might well be the case that the average of the 100 measurements would be under 10 ms, just as in our example.

However,

$$\mathbb{E}[X] = \sum_{i=0}^{\infty} i \Pr\{X = i\} = \sum_{i=1}^{\infty} i \left(\frac{1}{i} - \frac{1}{i+1} \right) = \sum_{i=1}^{\infty} \frac{1}{i+1} = \infty. \tag{19.73}$$

It is true that most of the time, the value of X will be small. But sometimes X will be very large and this happens with sufficient probability that the expected value of X is unbounded. In fact, if you keep repeating the experiment, you are likely to see some outcomes and averages that are much larger than 10 ms. In practice, such "outliers" are sometimes discarded, which masks the true behavior of X.

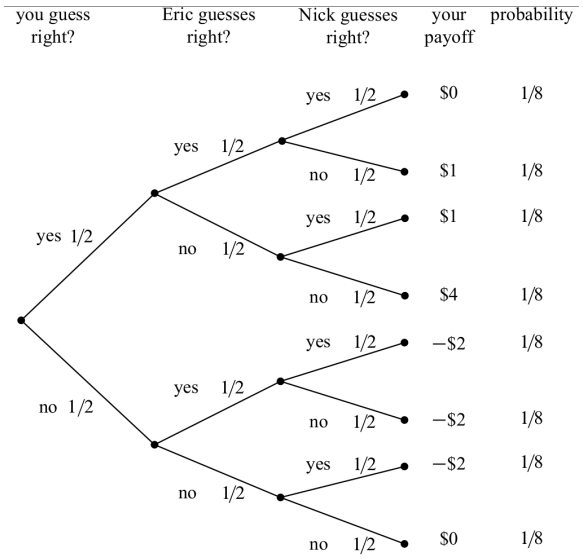

Figure 19.3
The tree diagram for the game where three players each wager $2 and then guess the outcome of a fair coin toss. The winners split the pot.

In general, the best way to compute an expected value in practice is to first use the experimental data to figure out the distribution as best you can, and then to compute its expectation. This method will help you identify cases where the expectation is infinite, and will generally be more accurate than a simple averaging of the data.

19.3.3 Expected Returns in Gambling Games

Each player will put $2 on the bar and secretly write "heads" or "tails" on their napkin. Then you will flip a fair coin. The $6 on the bar will then be "split", that is, be divided equally among the players who correctly predicted the outcome of the coin toss.

By using the four-step method, see Fig. 19.3, the expected return is

$$\mathbb{E}\left[\text{payoff}\right] = \frac{1}{8}\,(0+1+1+4+(-2)+(-2)+(-2)+0) = 0\,. \tag{19.74}$$

The "something" that is fishy is the opportunity that the other two players have to collude against you. One always guessed "tails" when the other always guessed "heads", and vice-versa. Since they are always guessing differently, one of them will always get a share of the pot. Therefore

$$\mathbb{E}\left[\text{payoff}\right] = \frac{1}{4}(0+1+1+4+(-2)+(-2)+(-2)+0) = -\frac{1}{2}\,. \tag{19.75}$$

In some cases, the collusion is inadvertent and you can profit from it. For example, many years ago, a former MIT Professor of Mathematics named Herman Chernoff figured out a way to make money by playing the state lottery. In a typical state lottery, all players pay $1 to play and select 4 numbers from 1 to 36, then the state draws 4 numbers from 1 to 36 uniformly at random. Finally, the states divides 1/2 of the money collected among the people who guessed correctly and spends the other half redecorating the governor's residence.

Chernoff discovered that a small set of numbers was selected by a large fraction of the population. The result is as though the players were intentionally colluding to lose. If any one of them guessed correctly, then they would have to split the pot with many other players. By selecting numbers uniformly at random, if he won, he would likely get the whole pot. His expected return was not -$0.5 as you might think, but $0.07.

19.3.4 Conditional Expectation

DEFINITION 19.36 **Conditional Expectation** The conditional expectation of a random variable X given event A is

$$\mathbb{E}[X \mid A] := \sum_{x \in \operatorname{ran} X} x \cdot \Pr\{X = x \mid A\}. \tag{19.76}$$

THEOREM 19.37 **Law of Total Expectation** Let X be a random variable on a sample space , and suppose that $A_1, A_2, \ldots$ is a partition of . Then

$$\mathbb{E}[X] = \sum_i \mathbb{E}[X \mid A_i] \Pr\{A_i\}. \tag{19.77}$$

Proof.

$$\begin{aligned}
\mathbb{E}[X] &= \sum_{x \in \operatorname{ran} X} x \cdot \Pr\{X = x\} && (19.78)\\
&= \sum_{x \in \operatorname{ran} X} x \sum_i \Pr\{X = x \mid A_i\} \Pr\{A_i\} && (19.79)\\
&= \sum_{x \in \operatorname{ran} X} \sum_i x \cdot \Pr\{X = x \mid A_i\} \Pr\{A_i\} && (19.80)\\
&= \sum_i \Pr\{A_i\} \sum_{x \in \operatorname{ran} X} x \cdot \Pr\{X = x \mid A_i\} && (19.81)\\
&= \sum_i \mathbb{E}[X \mid A_i] \Pr\{A_i\}. && (19.82)
\end{aligned}$$

□

19.4 Operations of Expectation

19.4.1 Expectations of Sums

THEOREM 19.38 For any random variables X_1 and X_2,

$$\mathbb{E}[X_1 + X_2] = \mathbb{E}[X_1] + \mathbb{E}[X_2]\,. \tag{19.83}$$

Proof. Let $X := X_1 + X_2$,

$$\begin{aligned}
\mathbb{E}[X] &= \sum_{\omega\in} X(\omega)\Pr\{\omega\} && (19.84)\\
&= \sum_{\omega\in} (X_1(\omega) + X_2(\omega)\Pr\{\omega\} && (19.85)\\
&= \sum_{\omega\in} X_1(\omega)\Pr\{\omega\} + \sum_{\omega\in} X_2(\omega)\Pr\{\omega\} && (19.86)\\
&= \mathbb{E}[X_1] + \mathbb{E}[X_2]\,. && (19.87)
\end{aligned}$$

□

THEOREM 19.39 For any random variables X_1 and X_2, and constants $a_1, a_2 \in \mathbb{R}$,

$$\mathbb{E}[a_1X_1 + a_2X_2] = a_1\,\mathbb{E}[X_1] + a_2\,\mathbb{E}[X_2]\,. \tag{19.88}$$

THEOREM 19.40 **Linearity of Expectation** For any random variables $X_1, X_2, \ldots, X_k$ and constants $a_1, a_2, \ldots, a_k \in \mathbb{R}$,

$$\mathbb{E}\left[\sum_{i=1}^{k} a_iX_i\right] = \sum_{i=1}^{k} a_i\,\mathbb{E}[X_i]\,. \tag{19.89}$$

In other words, expectation is a linear function.

Proof. By induction on k. □

The great thing about linearity of expectation is that no independence is required.

THEOREM 19.41 **Sums of Indicator Random Variables** Given any collection of events $A_1, A_2, \ldots, A_n$, the expected number of events that will occur is

$$\sum_{i=1}^{n} \Pr\{A_i\}\,. \tag{19.90}$$

Proof. Let

$$I := \sum_{i=1}^{n} I_{A_i}. \tag{19.91}$$

Therefore

$$\mathbb{E}[I] = \sum_{i=1}^{n} \mathbb{E}\left[I_{A_i}\right] = \sum_{i=1}^{n} \Pr\{A_i\}. \tag{19.92}$$

□

THEOREM 19.42 Given any collection of events $A_1, A_2, \ldots, A_n$, let $I := \sum_{i=1}^{n} I_{A_i}$, then

$$\Pr\{I \geq 1\} \leq \sum_{i=1}^{n} \Pr\{A_i\}. \tag{19.93}$$

19.4.2 The Coupon Collector Problem

THEOREM 19.43 Suppose there are n kinds of coupons, we can get one coupon uniformly and independently at random after each meal. The expected number of meals that we must purchase in order to acquire at least one of each kind of coupon is nH_n, where H_n is the n-th Harmonic number.

Proof. We partition the purchase sequence into n segments: $X_0, X_1, \ldots, X_{n-1}$, with each segment ends whenever we get a new kind of coupon. The total number of meals we must purchase to get all n kinds of coupons is the sum of the lengths of all these segments

$$X = \sum_{i=0}^{n-1} X_i. \tag{19.94}$$

For X_i, at the beginning of the i-th segment, we have i different kinds of coupons, and the segment ends when we acquire a new type. When we own i types, each meal contains a type that we already have with probability $\frac{i}{n}$, then each meal contains a new type with probability $1 - \frac{i}{n} = \frac{n-i}{n}$. Thus, the expected number of meals until we get a new kind of car is $\frac{n}{n-i}$ by the Mean Time to Failure rule. Therefore,

$$\mathbb{E}[X] = \mathbb{E}\left[\sum_{i=0}^{n-1} X_i\right] = \sum_{i=0}^{n-1} \mathbb{E}[X_i] = \sum_{i=0}^{n-1} \frac{n}{n-i} = n \sum_{i=1}^{n} \frac{1}{i} = nH_n \sim n \log n. \tag{19.95}$$

□

19.4.3 Bet Doubling Strategy

THEOREM 19.44 For the roulette game, the probability of winning of each time is 0.5, and the payoff for a bet on red or black matches the bet: for example, if you bet x dollars on red and the ball lands in a red slot, you get back your original x dollars bet plus another matching x. The bet doubling strategy is to bet red with x dollars and keep doubling until a red comes up. If your have an infinite amount of money to bet with, the expected return is x; if you only have a finite amount of money to bet with, the expected return is 0.

Proof. **Case 1**: If you have an infinite amount of money to bet with, red will *eventually* occur, say on the n-th bet. Then your return will be

$$2 \cdot 2^{n-1}x - \sum_{i=0}^{n-1} 2^i x = x\left(2^n - \frac{2^n - 1}{2-1}\right) = x\,. \tag{19.96}$$

Case 2: If you only have a finite amount of money to bet, say you can only afford n bets. Let $X_i :=$ the number of dollars you win on your i-th bet, where X_i is defined to be zero if red comes up before the i-th spin of the wheel. Since the wheel is fair,

$$\mathbb{E}[X_i] = 0.5 \cdot (2^{i-1}x) + 0.5 \cdot (-2^{i-1}x) = 0\,. \tag{19.97}$$

Therefore,

$$\mathbb{E}[X] = \mathbb{E}\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} \mathbb{E}[X_i] = 0\,. \tag{19.98}$$

□

19.4.4 Expectations of Products

LEMMA 19.45 For any two independent random variables X_1, X_2,

$$\mathbb{E}[X_1 \cdot X_2] = \mathbb{E}[X_1] \cdot \mathbb{E}[X_2]\,. \tag{19.99}$$

Proof.

$$\mathbb{E}[X_1 \cdot X_2] = \sum_{x \in \operatorname{ran} X_1 \cdot X_2} x \Pr\{X_1 \cdot X_2 = x\} \tag{19.100}$$

$$= \sum_{x_1 \in \operatorname{ran} X_1} \sum_{x_2 \in \operatorname{ran} X_2} x_1 x_2 \Pr\{X_1 = x_1 \wedge X_2 = x_2\} \tag{19.101}$$

$$= \sum_{x_1 \in \operatorname{ran} X_1} \sum_{x_2 \in \operatorname{ran} X_2} x_1 x_2 \Pr\{X_1 = x_1\} \Pr\{X_2 = x_2\} \tag{19.102}$$

$$= \left(\sum_{x_1 \in \operatorname{ran} X_1} x_1 \Pr\{X_1 = x_1\} \right) \left(\sum_{x_2 \in \operatorname{ran} X_2} x_2 \Pr\{X_2 = x_2\} \right) \quad (19.103)$$

$$= \mathbb{E}[X_1] \cdot \mathbb{E}[X_2] \,. \quad (19.104)$$

□

THEOREM 19.46 **Expectation of Independent Product** If random variables $X_1, X_2, \ldots, X_n$ are mutually independent, then

$$\mathbb{E}\left[\prod_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} \mathbb{E}[X_i] \,. \quad (19.105)$$

19.4.5 Expectations of Quotients

THEOREM 19.47 Usually,

$$\mathbb{E}\left[\frac{1}{X}\right] \neq \frac{1}{\mathbb{E}[X]} \,. \quad (19.106)$$

THEOREM 19.48 Usually,

$$\mathbb{E}\left[\frac{X}{Y}\right] \neq \frac{\mathbb{E}[X]}{\mathbb{E}[Y]} \,. \quad (19.107)$$

19.5 Expectations of Different Random Variables

19.5.1 Expectation of an Indicator Random Variable

THEOREM 19.49 **Expectation of an Indicator Random Variable** If I_A is the indicator random variable for event A, then

$$\mathbb{E}[I_A] = \Pr\{A\} \,. \quad (19.108)$$

Proof.

$$\mathbb{E}[I_A] = 1 \cdot \Pr\{I_A = 1\} + 0 \cdot \Pr\{I_A = 0\} = \Pr\{I_A = 1\} = \Pr\{A\} \,. \quad (19.109)$$

□

19.5.2 Expectation of a Uniform Random Variable

THEOREM 19.50 **Expectation of a Uniform Random Variable** Suppose X is a random variable has a uniform distribution on $[1, n]$, then

$$\mathbb{E}[X] = \frac{n+1}{2}. \tag{19.110}$$

Proof.

$$\mathbb{E}[X] = \frac{1}{n}\sum_{i=1}^{n} i = \frac{1}{n}\frac{n(n+1)}{2} = \frac{n+1}{2}. \tag{19.111}$$

□

19.5.3 Expectation of a Geometric Random Variable

THEOREM 19.51 **Expectation of a Geometric Random Variable** Suppose X is a random variable has a geometric distribution with parameter p, then

$$\mathbb{E}[X] = \frac{1}{p}. \tag{19.112}$$

Proof. Let $A :=$ the event that $X = 1$, i.e., the system fails on the first step. By the Law of Total Expectation,

$$\mathbb{E}[X] = \mathbb{E}[X \mid A]\Pr\{A\} + \mathbb{E}\left[X \mid \bar{A}\right]\Pr\{\bar{A}\}. \tag{19.113}$$

Since A is the condition that the system crashes on the first step,

$$\mathbb{E}[X \mid A] = 1. \tag{19.114}$$

Since $\bar{A}$ is the condition that the system does not crash on the first step, conditioning on $\bar{A}$ is equivalent to taking a first step without failure and then starting over without conditioning. Hence,

$$\mathbb{E}\left[X \mid \bar{A}\right] = 1 + \mathbb{E}[X]. \tag{19.115}$$

Therefore

$$\mathbb{E}[X] = 1 \cdot p + (1 + \mathbb{E}[X])(1-p) = 1 + (1-p)\,\mathbb{E}[X]. \tag{19.116}$$

$$\mathbb{E}[X] = \frac{1}{p}. \tag{19.117}$$

□

19.5.4 Expectation of a Binomial Random Variable

THEOREM 19.52 **Expectation of a Binomial Random Variable** Suppose X is a random variable has a binomial distribution with parameter n and p, then

$$\mathbb{E}[X] = np\,. \tag{19.118}$$

Proof. Express X as a sum of indicator random variables:

$$X = \sum_{i=1}^{n} X_i\,, \tag{19.119}$$

where $\forall i.\ X_i \sim Ber(p)$. Therefore

$$\mathbb{E}[X] = \sum_{i=1}^{n} \mathbb{E}[X_i] = \sum_{i=1}^{n} \Pr\{X_i = 1\} = np\,. \tag{19.120}$$

□

20 Deviation From the Mean

20.1 Variance

20.1.1 Definitions

DEFINITION 20.1 **Variance/Mean Square Deviation σ^2** The variance of a random variable X is the expectation of the square of the amount by which X differs from its expectation.

$$\sigma^2 := \operatorname{var} X := \mathbb{E}\left[(X - \mathbb{E}[X])^2\right] = \mathbb{E}\left[(X-\mu)^2\right]. \tag{20.1}$$

DEFINITION 20.2 **Standard Deviation/Root Mean Square σ** The standard deviation of a random variable X is the square root of the variance

$$\sigma := \sqrt{\operatorname{var} X} = \sqrt{\mathbb{E}\left[(X - \mathbb{E}[X])^2\right]} = \sqrt{\mathbb{E}\left[(X-\mu)^2\right]}. \tag{20.2}$$

LEMMA 20.3 For any random variable X,

$$\sigma^2 = \mathbb{E}\left[X^2\right] - \mu^2. \tag{20.3}$$

Proof.

$$\sigma^2 = \mathbb{E}\left[(X-\mu)^2\right] = \mathbb{E}\left[X^2 - 2\mu X + \mu^2\right] = \mathbb{E}\left[X^2\right] - 2\mu\,\mathbb{E}[X] + \mu^2 = \mathbb{E}\left[X^2\right] - \mu^2 \tag{20.4}$$

□

COROLLARY 20.4 For any random variable X,

$$\mathbb{E}\left[X^2\right] \geq \mu^2. \tag{20.5}$$

The equality holds true exactly when X is a constant.

Proof.

$$\sigma^2 = \mathbb{E}\left[X^2\right] - \mu^2 \geq 0. \tag{20.6}$$

$$\mathbb{E}\left[X^2\right] = \mu^2 \Leftrightarrow \sigma^2 = 0 \Leftrightarrow \Pr\{X = \mu\} = 1. \tag{20.7}$$

□

20.1.2 Properties of Variances

THEOREM 20.5 Let X be a random variable, and a. b be constants. Then

$$\operatorname{var}(aX+b) = a^2 \operatorname{var} X. \tag{20.8}$$

Proof.

$$\mathbb{E}[aX+b] = a\,\mathbb{E}[X] + b. \tag{20.9}$$

$$\begin{aligned}
\operatorname{var}(aX+b) &= \mathbb{E}\left[(aX+b-\mathbb{E}[aX+b])^2\right] && (20.10)\\
&= \mathbb{E}\left[(aX-\mathbb{E}[X])^2\right] && (20.11)\\
&= \mathbb{E}\left[a^2(X-\mathbb{E}[X])\right] && (20.12)\\
&= a^2\,\mathbb{E}\left[(X-\mathbb{E}[X])^2\right] && (20.13)\\
&= a^2 \operatorname{var} X. && (20.14)
\end{aligned}$$

□

THEOREM 20.6 If X_1 and X_2 are independent random variables, then

$$\operatorname{var}(X_1+X_2) = \operatorname{var} X_1 + \operatorname{var} X_2. \tag{20.15}$$

Proof.

$$\begin{aligned}
\operatorname{var}(X_1+X_2) &= \mathbb{E}\left[(X_1+X_2)^2\right] - \mathbb{E}[X_1+X_2]^2 && (20.16)\\
&= \mathbb{E}\left[X_1^2\right] + \mathbb{E}\left[X_2^2\right] + 2\,\mathbb{E}[X_1X_2] - (\mathbb{E}[X_1]+\mathbb{E}[X_2])^2 && (20.17)\\
&= \mathbb{E}\left[X_1^2\right] - \mathbb{E}[X_1]^2 + \mathbb{E}\left[X_2^2\right] - \mathbb{E}[X_2]^2 + 2(\mathbb{E}[X_1X_2] - \mathbb{E}[X_1]\,\mathbb{E}[X_2]) && (20.18)\\
&= \operatorname{var} X_1 + \operatorname{var} X_2. && (20.19)
\end{aligned}$$

□

THEOREM 20.7 **Pairwise Independent Additivity of Variance** If $X_1, X_2, \ldots, X_n$ are pairwise independent random variables, then

$$\operatorname{var} \sum_{i=1}^{n} X_i = \sum_{i=1}^{n} \operatorname{var} X_i. \tag{20.20}$$

20.1.3 Variances of Different Random Variables

THEOREM 20.8 **Variance of an Indicator Random Variable** If I_A is the indicator random variable for event A, then

$$\operatorname{var} I_A = \Pr\{A\}(1 - \Pr\{A\}). \tag{20.21}$$

Proof.

$$\operatorname{var} I_A = \mathbb{E}\left[I_A^2\right] - \mathbb{E}[I_A]^2 = \mathbb{E}[I_A] - \mathbb{E}[I_A]^2 = \Pr\{A\} - \Pr\{A\}^2 = \Pr\{A\}(1 - \Pr\{A\}). \tag{20.22}$$

□

THEOREM 20.9 **Variance of a Uniform Random Variable** Suppose X is a random variable has a uniform distribution on $[1, n]$, then

$$\operatorname{var} X = \frac{n^2 - 1}{12}. \tag{20.23}$$

Proof.

$$\mathbb{E}\left[X^2\right] = \frac{1}{n}\sum_{i=1}^{n} i^2 = \frac{1}{n}\frac{(2n+1)n(n+1)}{6} = \frac{(2n+1)(n+1)}{6}. \tag{20.24}$$

$$\operatorname{var} X = \mathbb{E}\left[X^2\right] - \mathbb{E}[X]^2 = \frac{(2n+1)(n+1)}{6} - \left(\frac{n+1}{2}\right)^2 = \frac{n^2-1}{2}. \tag{20.25}$$

□

THEOREM 20.10 **Variance of a Geometric Random Variable** Suppose X is a random variable has a geometric distribution with parameter p, then

$$\operatorname{var} X = \frac{1-p}{p^2}. \tag{20.26}$$

Proof. Let $A :=$ the event that $X = 1$, i.e., the system fails on the first step. By the Law of Total Expectation,

$$\mathbb{E}\left[X^2\right] = \mathbb{E}\left[X^2 \mid A\right]\Pr\{A\} + \mathbb{E}\left[X^2 \mid \bar{A}\right]\Pr\{\bar{A}\}. \tag{20.27}$$

Since A is the condition that the system crashes on the first step,

$$\mathbb{E}\left[X^2 \mid A\right] = 1. \tag{20.28}$$

Since $\bar{A}$ is the condition that the system does not crash on the first step, conditioning on $\bar{A}$ is equivalent to taking a first step without failure and then starting over without conditioning. Hence,

$$\mathbb{E}\left[X^2 \mid \bar{A}\right] = \mathbb{E}\left[(1+X)^2\right]. \tag{20.29}$$

Therefore

$$\begin{aligned}
\mathbb{E}\left[X^2\right] &= 1 \cdot p + \mathbb{E}\left[(1+X)^2\right](1-p) && (20.30)\\
&= p + \mathbb{E}\left[X^2\right](1-p) + 2\,\mathbb{E}[X](1-p) + (1-p) && (20.31)\\
&= 1 + \mathbb{E}\left[X^2\right](1-p) + 2\frac{1-p}{p}. && (20.32)
\end{aligned}$$

$$\mathbb{E}\left[X^2\right] = \frac{2-p}{p^2}. \tag{20.33}$$

$$\operatorname{var} X = \mathbb{E}\left[X^2\right] - \mathbb{E}[X] = \frac{2-p}{p^2} - \frac{1}{p^2} = \frac{1-p}{p^2}. \tag{20.34}$$

□

THEOREM 20.11 **Variance of a Binomial Random Variable** Suppose X is a random variable has a binomial distribution with parameter n and p, then

$$\operatorname{var} X = np(1-p). \tag{20.35}$$

Proof. Express X as a sum of indicator random variables:

$$X = \sum_{i=1}^{n} X_i, \tag{20.36}$$

where $\forall i.\ X_i \sim Ber(p)$. Therefore

$$\operatorname{var} X = \sum_{i=1}^{n} \operatorname{var} X_i = \sum_{i=1}^{n} p(1-p) = np(1-p). \tag{20.37}$$

□

20.2 Probabilistic Bounds

The more you know about a random variable, the better bounds you can derive on the probability that it derives from its mean.

20.2.1 Markov's Theorem

THEOREM 20.12 **Markov's Theorem** If X is a nonnegative random variable, and $\mu = \mathbb{E}[X]$, then

$$\forall c > 0.\ \Pr\{X \geq c\} \leq \frac{\mu}{c} = O\left(\frac{1}{c}\right). \tag{20.38}$$

Proof.

$$\begin{aligned}
\mathbb{E}[X] &= \sum_x x \Pr\{X = x\} && (20.39)\\
&\geq \sum_{x \geq c} x \Pr\{X = x\} \quad \text{// since } X \geq 0 && (20.40)\\
&\geq \sum_{x \geq c} c \Pr\{X = x\} && (20.41)\\
&= c \sum_{x \geq c} \Pr\{X = x\} && (20.42)\\
&= c \Pr\{X \geq c\}. && (20.43)
\end{aligned}$$

□

COROLLARY 20.13 If X is a nonnegative random variable, and $\mu = \mathbb{E}[X]$, then

$$\forall c \geq 1.\ \Pr\{X \geq c\mu\} \leq \frac{1}{c}. \tag{20.44}$$

Proof. By substituting c from Markov's Theorem to $c\mu$. □

COROLLARY 20.14 Let X be a random variable such that $X \geq x_0$ for some $x_0 \in \mathbb{R}$, and $\mu = \mathbb{E}[X]$, then

$$\forall c \geq x_0.\ \Pr\{X \geq c\} \leq \frac{\mu - x_0}{c - x_0}. \tag{20.45}$$

Proof. Let $X' := X - x_0$, then X' is a nonnegative random variable with mean

$$\mathbb{E}\left[X'\right] = \mathbb{E}[X - x_0] = \mu - x_0. \tag{20.46}$$

Hence, Markov's Theorem implies that

$$\Pr\{X \geq c\} = \Pr\{X' - x_0 \geq c\} = \Pr\{X' \geq c - x_0\} \leq \frac{\mathbb{E}\left[X'\right]}{c - x_0} = \frac{\mu - x_0}{c - x_0}. \tag{20.47}$$

□

COROLLARY 20.15 Let X be a random variable such that $X \leq x_0$ for some $x_0 \in \mathbb{R}$, and $\mu = \mathbb{E}[X]$, then

$$\forall c \leq x_0.\ \Pr\{X \leq c\} \leq \frac{x_0 - \mu}{x_0 - c}. \tag{20.48}$$

Proof. Let $X' := x_0 - X$, then X' is a nonnegative random variable with mean

$$\mathbb{E}\left[X'\right] = \mathbb{E}[x_0 - X] = x_0 - \mu. \tag{20.49}$$

Hence, Markov's Theorem implies that

$$\Pr\{X \geq c\} = \Pr\{x_0 - X' \geq c\} = \Pr\{X' \leq x_0 - c\} \leq \frac{\mathbb{E}\left[X'\right]}{x_0 - c} = \frac{x_0 - \mu}{x_0 - c}. \tag{20.50}$$

□

20.2.2 Chebyshev's Theorem

THEOREM 20.16 **Chebyshev's Theorem** For any random variable X,

$$\forall c > 0.\ \Pr\{|X - \mu| \geq c\} \leq \frac{\sigma^2}{c^2} = O\left(\frac{1}{c^2}\right). \tag{20.51}$$

Proof.

$$\Pr\{|X - \mu| \geq c\} = \Pr\{(X - \mu)^2 \geq c^2\} \leq \frac{\mathbb{E}\left[(X - \mu)^2\right]}{c^2} = \frac{\sigma^2}{c^2}. \tag{20.52}$$

□

COROLLARY 20.17 For any random variable X,

$$\forall c > 0.\ \Pr\{|X - \mu| \geq c\sigma\} \leq \frac{1}{c^2}. \tag{20.53}$$

Proof. By substituting c from Chebyshev's Theorem to $c\sigma$. □

THEOREM 20.18 **Pairwise Independent Sampling** Let $X_1, X_2, \ldots, X_n$ be pairwise independent variables with mean μ and variance σ^2. Let the sample mean $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} X_i$, then

$$\Pr\{|\hat{\mu} - \mu| \geq c\} \leq \frac{\sigma^2}{c^2 n}. \tag{20.54}$$

Proof.

$$\mathbb{E}\left[\hat{\mu}\right] = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} X_i\right] = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}[X_i] = \frac{n\mu}{\mu} = \mu. \tag{20.55}$$

$$\operatorname{var}\hat{\mu} = \operatorname{var}\frac{1}{n}\sum_{i=1}^{n} X_i = \frac{1}{n^2}\sum_{i=1}^{n}\operatorname{var} X_i = \frac{n\sigma^2}{n^2} = \frac{\sigma^2}{n}. \tag{20.56}$$

By Chebyshev's Theorem,

$$\Pr\{|\hat{\mu} - \mu| \geq c\} \leq \frac{\operatorname{var}\hat{\mu}}{c^2} = \frac{\sigma^2}{c^2 n}. \tag{20.57}$$

□

COROLLARY 20.19 **Weak Law of Large Numbers** Let $X_1, X_2, \ldots, X_n$ be pairwise independent variables with mean μ and finite variance σ^2. Let the sample mean $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} X_i$, then

$$\lim_{n\to\infty} \Pr\{|\hat{\mu} - \mu| \geq c\} = 0. \tag{20.58}$$

Proof.

$$\lim_{n\to\infty} \Pr\{|\hat{\mu} - \mu| \geq c\} \leq \lim_{n\to\infty} \frac{\sigma^2}{c^2 n} = 0. \tag{20.59}$$

□

20.3 Mutually Independent Random Variables

If all you know about a random variable is its mean and variance, then Chebyshev's Theorem is the best you can do to bound the probability. However, if the random variable X is a sum of n mutually independent random variables, a tighter bound can be derived.

20.3.1 The Chernoff Bound

THEOREM 20.20 **Chernoff Bound** Let $X_1, X_2, \ldots, X_n$ be mutually independent random variables such that $\forall i.\ 0 \leq X_i \leq 1$. Let $X = \sum_{i=1}^{n} X_i$, and $\mu = \mathbb{E}[X]$. Then

$$\forall c \geq 1.\ \Pr\{X \geq c\mu\} \leq \frac{1}{\exp(\beta(c)\mu)} = O\left(\frac{1}{\exp c}\right), \tag{20.60}$$

where

$$\beta(c) := c\log c - c + 1. \tag{20.61}$$

In other words, the sum of lots of little, independent random variables is unlikely to significantly exceed the mean of the sum.

20.3.2 Randomized Load Balancing

Say there are n tasks in every 10 minutes, each task is assigned to one of the k computers for processing, and each computer works sequentially through its assigned tasks. Processing an average task tasks a computer 1/4 seconds, but the most protracted harangues require 1 second.

If the length of every task were known in advance, then finding a balanced distribution would be a kind of "bin packing" problem. Such problems are hard to solve exactly, though approximation algorithms can come close. However, in this case, task length are not known in advance, but it turns out that random assignment works reasonably well.

To begin, let's find the probability that the first server is overloaded. Let $X_i :=$ number the of seconds that the first server spends on the i-th task. So $X_i = 0$ if the task is assigned to another machine; otherwise X_i is the length of the task. Then $X = \sum_{i=1}^{n} X_i$ is the total length of tasks assigned to the server. We want to compute $\Pr\{X \geq 600\}$, that is, the first server is assigned more than 10 minutes of work.

Since there are n tasks with an expected length of $1/4$ second, and tasks are assigned to k computers at random. The expected load on the first server is

$$\mu = \mathbb{E}[X] = \frac{1/4 \times n}{k} = \frac{n}{4k}. \tag{20.62}$$

Assume that task lengths and assignments are independent. Suppose $n = 24,000$, by Chernoff's bound,

$$\Pr\{X \geq 600\} = \Pr\left\{X \geq \frac{k}{10}\mu\right\} \leq \frac{1}{\exp(-\beta(c)\mu)}, \tag{20.63}$$

where $c = \frac{k}{10}$. Then by Union Bound,

$$\Pr\{\text{some server is overloaded}\} \leq \sum_{j=1}^{k} \Pr\{\text{server } j \text{ is overloaded}\} \leq \frac{k}{\exp(-\beta(c)\mu)} \tag{20.64}$$

When $k = 13$, that bound is 0.0000000760.

20.3.3 Murphy's Law

Murphy's Law[1] says that if a random variable is an independent sum of 0/1-valued variables and has a large expectation, then there is a huge probability of getting a value of at least 1.

1 This is in reference and deference to the famous saying that "If something can go wrong, it probably will."

THEOREM 20.21 **Murphy's Law** Let $A_1, A_2, \ldots, A_n$ be mutually independent events. Let $I := \sum_{i=1}^{n} I_{A_i}$ be the number of events that occur. Then

$$\Pr\{I = 0\} \leq \frac{1}{\exp \mathbb{E}[I]}. \tag{20.65}$$

Proof.

$$\begin{aligned}
\Pr\{I = 0\} &= \Pr\left\{\bigcap_{i=1}^{n} \bar{A}_i\right\} && (20.66)\\
&= \prod_{i=1}^{n} \Pr\{\bar{A}_i\} && (20.67)\\
&= \prod_{i=1}^{n} (1 - \Pr\{A_i\}) && (20.68)\\
&\leq \prod_{i=1}^{n} \exp(-\Pr\{A_i\}) \quad (\text{since } 1 - x \leq \exp(-x)) && (20.69)\\
&= \exp\left(-\sum_{i=1}^{n} \Pr\{A_i\}\right) && (20.70)\\
&= \exp\left(-\sum_{i=1}^{n} \mathbb{E}\left[I_{A_i}\right]\right) && (20.71)\\
&= \exp(-\mathbb{E}[I]). && (20.72)
\end{aligned}$$

□

This result can help to explain "coincidences", "miracles", and crazy events that seem to have been very unlikely to happen. Such events do happen, in part, because there are so many possible unlikely events that the sum of their probabilities is greater than one.

21 Random Walks

DEFINITION 21.1 **Random Walks** Model of situations in which an object moves in sequence of steps in randomly chosen directions.

DEFINITION 21.2 **Unbiased Random Walk** The value in the random walk is equally likely to move in each direction.

DEFINITION 21.3 **Boundary Condition/Absorbing Barrier** Situation when the walk ends when a certain value is reached.

21.1 Unbiased Random Walks

21.1.1 Death is Certain

THEOREM 21.4 For an unbiased, one-dimensional random walk with absorbing barriers at positions 0 and w. The walk begins at position n. Then

$$\Pr\{\text{the bug falls to the right}\} = \frac{n}{w}, \tag{21.1}$$

$$\Pr\{\text{the bug falls to the left}\} = \frac{w-n}{w}. \tag{21.2}$$

Proof. Let $R_n :=$ the probability that the bug falls to the right given that it starts at position n. If it starts at position w, he falls into the right immediately, so $R_w = 1$. On the other hand, if he stars at position 0, then he falls to the left immediately, so $R_0 = 0$. Now suppose that $0 < n < w$, we can break it up into two cases. Therefore

$$R_n = \begin{cases} 0 & \text{if } n = 0; \\ \frac{1}{2}R_{n-1} + \frac{1}{2}R_{n+1} & \text{if } 0 < n < w; \\ 1 & \text{if } n = w. \end{cases} \tag{21.3}$$

This is a linear recurrence. R_n is a function both a preceding term R_{n-1} and a following term R_{n+1}. We can rearrange the terms in the recurrence equation:

$$R_{n+1} = 2R_n - R_{n-1}. \tag{21.4}$$

The characteristic equation is

$$x^2 - 2x - 1 = 0. \tag{21.5}$$

This equation has double root $x = 1$, so the general solution has the form

$$R_n = a \cdot 1^n + b \cdot n \cdot 1^n = a + bn. \tag{21.6}$$

Substituting the boundary condition gives

$$R_n = \frac{n}{w}. \tag{21.7}$$

We can compute the probability that the bug falls off the left by exploiting the symmetry of the problem: the probability that it falls left side starting at position n is the same as the probability that it falls right side starting at position $w - n$, which is $\frac{w-n}{w}$.

Besides, since

$$\frac{n}{w} + \frac{w-n}{w} = 1, \tag{21.8}$$

the probability that the bug hops around forever is 0. □

COROLLARY 21.5 Suppose $w = \infty$, then the probability that the bug eventually falls into left is

$$\lim_{w\to\infty} \frac{w-n}{w} = 1. \tag{21.9}$$

21.1.2 Life Expectancy

THEOREM 21.6 For an unbiased, one-dimensional random walk with absorbing barriers at positions 0 and w. The walk begins at position n. Let $X :=$ number of hops the bug takes before falling off an edge. Then

$$\mathbb{E}[X] = n(w-n). \tag{21.10}$$

Proof. Let $X_n :=$ number of hops the bug takes before falling off an edge starting at position n. If it starts at either edge of the island, then it dies immediately: $X_0 = X_w = 0$. If $0 < n < w$, then we can again break down the analysis into two cases. Therefore,

$$X_n = \begin{cases} 0 & \text{if } n = 0; \\ 1 + \frac{1}{2}X_{n-1} + \frac{1}{2}X_{n+1} & \text{if } 0 < n < w; \\ 0 & \text{if } n = w. \end{cases} \tag{21.11}$$

We can rewrite it as

$$X_{n+1} = 2X_n - X_{n-1} - 2. \tag{21.12}$$

The characteristic equation is

$$x^2 - 2x + 1 = 0. \tag{21.13}$$

There is a double root $x = 1$, so the homogeneous solution has the form:

$$X_n = a + bn. \tag{21.14}$$

We also need to find a particular solution. Trying $X_n = c$ and $X_n = c + dn$ do not work, and trying $X_n = c + dn + en^2$ gives $X_n = -n^2$.

The general form of the solution is

$$X_n = a + bn - n^2 . \tag{21.15}$$

Substituting the boundary conditions gives

$$X_n = wn - n^2 = n(w - n) . \tag{21.16}$$

□

COROLLARY 21.7 When $w = \infty$, then the expected lifespan is

$$\lim_{w\to\infty} n(w - n) = \infty . \tag{21.17}$$

21.2 Biased Random Walks

21.2.1 Gambler's Ruin

THEOREM 21.8 For an biased, one-dimensional random walk with absorbing barriers at positions 0 and w. Each step, the bug goes right with probability p, and goes left with probability $1 - p$. The walk begins at position n. Then

$$\Pr\{\text{the bug falls to the right}\} = \frac{\left(\frac{1-p}{p}\right)^n - 1}{\left(\frac{1-p}{p}\right)^w - 1} . \tag{21.18}$$

Proof. Let $R_n :=$ the probability that the bug falls to the right given that it starts at position n. If it starts at position w, he falls into the right immediately, so $R_w = 1$. On the other hand, if he stars at position 0, then he falls to the left immediately, so $R_0 = 0$. Now suppose that $0 < n < w$, we can break it up into two cases. Therefore

$$R_n = \begin{cases} 0 & \text{if } n = 0; \\ (1-p)R_{n-1} + pR_{n+1} & \text{if } 0 < n < w; \\ 1 & \text{if } n = w. \end{cases} \tag{21.19}$$

This is a linear recurrence.

$$pR_{n+1} = R_n - (1-p)R_{n-1} . \tag{21.20}$$

The characteristic equation is

$$px^2 - x + (1-p) = 0 . \tag{21.21}$$

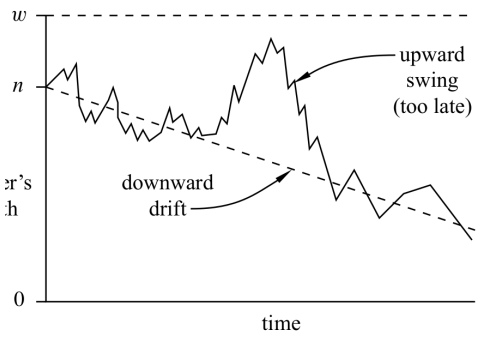

Figure 21.1
In a biased random walk, the downward drift usually dominates the swings of good luck over the long term.

This equation has roots $x_1 = \frac{1-p}{p}$, $x_2 = 1$. If $p \neq \frac{1}{2}$, so the general solution has the form

$$R_n = a \cdot \left(\frac{1-p}{p}\right)^n + b \cdot 1^n = a \cdot \left(\frac{1-p}{p}\right)^n + b. \tag{21.22}$$

Substituting the boundary condition gives

$$R_n = \frac{\left(\frac{1-p}{p}\right)^n - 1}{\left(\frac{1-p}{p}\right)^w - 1}. \tag{21.23}$$

□

Intuitively, there are two forces at work. First, the bug's position has random upward and downward *swings* due the runs of good and bad luck. Second, the position has a steady, downward *drift* because it has a small expected loss on every step. The problem is that its position is steadily drifting downward, and it needs a huge upward swing to save itself, but such a huge swing is extremely improbable. See Fig. 21.1.

21.2.2 Life Expectancy

THEOREM 21.9 For an biased, one-dimensional random walk with absorbing barriers at positions 0 and w. Each step, the bug goes right with probability p, and goes left with probability $1-p$. The walk begins at position n. Let $X :=$ number of hops the bug takes before falling off an edge. Then

$$\mathbb{E}[X] = \frac{n}{1-2p} - \frac{w}{1-2p} \frac{\left(\frac{1-p}{p}\right)^n - 1}{\left(\frac{1-p}{p}\right)^w - 1}. \tag{21.24}$$

Proof. Let $X_n :=$ number of hops the bug takes before falling off an edge starting at position n. If it starts at either edge of the island, then it dies immediately: $X_0 = X_w = 0$. If $0 < n < w$, then we can again break down the analysis into two cases. Therefore,

$$X_n = \begin{cases} 0 & \text{if } n = 0; \\ 1 + (1-p)X_{n-1} + pX_{n+1} & \text{if } 0 < n < w; \\ 0 & \text{if } n = w. \end{cases} \tag{21.25}$$

The characteristic equation is

$$px^2 - x + (1-p) = 0\,. \tag{21.26}$$

This equation has roots $x_1 = \frac{1-p}{p}$, $x_2 = 1$. If $p \neq \frac{1}{2}$, so the general solution has the form

$$X_n = a \cdot \left(\frac{1-p}{p}\right)^n + b \cdot 1^n = a \cdot \left(\frac{1-p}{p}\right)^n + b\,. \tag{21.27}$$

We also need to find a particular solution. Trying $X_n = c$ does not work, and trying $X_n = c + dn$ gives $X_n = \frac{n}{1-2p}$.

The general form of the solution is

$$X_n = a \cdot \left(\frac{1-p}{p}\right)^n + b + \frac{n}{1-2p}\,. \tag{21.28}$$

Substituting the boundary conditions gives

$$X_n = \frac{n}{1-2p} - \frac{w}{1-2p} \frac{\left(\frac{1-p}{p}\right)^n - 1}{\left(\frac{1-p}{p}\right)^w - 1}\,. \tag{21.29}$$

□

COROLLARY 21.10 Suppose $p < \frac{1}{2}$, and $w - n$ is large, then

$$\lim_{(w-n)\to\infty} \left(\frac{n}{1-2p} - \frac{w}{1-2p} \frac{\left(\frac{1-p}{p}\right)^n - 1}{\left(\frac{1-p}{p}\right)^w - 1} \right) = \frac{n}{1-2p}\,. \tag{21.30}$$

Proof.

$$\lim_{(w-n)\to\infty} \left(\frac{n}{1-2p} - \frac{w}{1-2p} \frac{\left(\frac{1-p}{p}\right)^n - 1}{\left(\frac{1-p}{p}\right)^w - 1} \right) \tag{21.31}$$

$$= \lim_{(w-n)\to\infty} \frac{n}{1-2p} - \lim_{(w-n)\to\infty} \frac{w}{1-2p} \left(\frac{1-p}{p}\right)^{-(w-n)} \tag{21.32}$$

$$= \frac{n}{1-2p}\,. \tag{21.33}$$

□

21.3 Random Walks on Graphs

Traditional document searching programs give scores for each document based on the frequency or position that the search terms appeared in the document. However, there might be websites filled with repeat certain words over and over in order to attract visitors. Google's enormous market capital in part derives from the revenue it receives from advertisers paying to appear at the top of search results. That top placement would not be worth much if Google's results were as easy to manipulate as keyword frequencies.

21.3.1 A First Crack at Page Rank

The hyperlink structure of the World Wide Web can be described as a digraph. The vertices are the web pages with a directed edge from vertex x to vertex y if x has a link to y.

Their first idea: try defining the page rank of x to be $\text{indeg}\, x$, the number of links pointing to x. The idea is to think of web pages as voting for the most important page—the more votes, the better the rank. Unfortunately, there are some problems with this idea. One thing you could do to have your page get a high ranking is to create lots of dummy pages with links to your page. There is another problem: a page could become unfairly influential by having lots of links to other pages it wanted to hype.

21.3.2 Random Walk on the Web Graph

Instead of just counting the indegree of a vertex, they considered the probability of being at each page after a long random walk on the web graph. In particular, they decided to model a user's web experience as following each link on a page with uniform probability. More generally, they assigned each edge $x \to y$ of the web graph with a probability conditioned on being on page x:

$$\Pr\{\text{follow link } x \to y \mid \text{at page } x\} := \frac{1}{\text{outdeg}\, x}. \tag{21.34}$$

Suppose each vertex is assigned a probability that corresponds, intuitively, to the likelihood that a random walker is at that vertex at a randomly chosen time. We assume that the walk never leaves the vertices in the graph, so we require that

$$\sum_x \Pr\{\text{at } x\} = 1. \tag{21.35}$$

We can also compute the probability of arriving at a particular page y by summing over all edges pointing to y.

$$\Pr\{\text{go to } y\} = \sum_{x \to y} \Pr\{\text{follow link } x \to y \mid \text{at page } x\} \Pr\{\text{at page } x\} = \sum_{x \to y} \frac{\Pr\{\text{at page } x\}}{\text{outdeg}\, x} \tag{21.36}$$

However, some pages have no hyperlinks out. Under the current model, the user cannot escape these pages. In reality, however, the user does not fall off the end of the web into a void of nothingness. Instead, he restarts his web journey. Moreover, even if a user does not get stuck at a dead end, they will commonly get discouraged after following some unproductive path for a while and will decide to restart.

To model this aspect of the web, Sergey and Larry added a *supervertex* to the web graph and added an edge from every page to the supervertex. Moreover, the supervertex points to every other vertex in the graph with equal probability, allowing the walk to restart from a random place. This ensures that the graph is strongly connected.

If a page had no hyperlinks, then its edge to the supervertex has to be assigned probability one. For pages that had some hyperlinks, the additional edge pointing to the supervertex was assigned some specially given probability. In the original versions of Page Rank, this probability was arbitrarily set to 0.15; its other $\operatorname{outdeg} x$ outgoing edges were still kept equally likely.

21.3.3 Stationary Distribution and Page Rank

DEFINITION 21.11 **Stationary Distribution** An assignment of probabilities to vertices in a digraph is a stationary distribution if for all vertices x

$$\Pr\{\text{at } x\} = \Pr\{\text{go to } x\}\,. \tag{21.37}$$

The basic idea behind page rank is finding a stationary distribution over the web graph. This is been done by solving following linear equations:

$$\Pr\{\text{at } y\} = \sum_{x \to y} \frac{\Pr\{\text{at page } x\}}{\operatorname{outdeg} x}\,. \tag{21.38}$$

$$\sum_{x} \Pr\{\text{at } x\} = 1\,. \tag{21.39}$$

Strongly connected graphs have unique stationary distributions, and their addition of a supervertex ensures this. Note that general digraphs without supervertices may have neither of these properties: there may not be a unique stationary distribution, and even when there is, there may be starting points from which the probabilities of positions during a random walk do not onverge to the stationary distribution.