

Notes on *Essentials*
of Discrete Mathematics

Michael T. Wescoat

January 10, 2016

Chapter 1

Logical Thinking

1.1 Formal Logic

1.1.1 Inquiry Problems

Foo

1.1.2 Connectives and Propositions

Name	Symbol
and	\wedge
or	\vee
not	\neg
implies (if...then)	\rightarrow
if and only if	\leftrightarrow

1.1.3 Truth Tables

Foo

1.1.4 Logical Equivalences

Foo

1.2 Propositional Logic

1.2.1 Tautologies and Contradictions

Foo

1.2.2 Derivation Rules

Foo

1.2.3 Proof Sequences

Foo

1.2.4 Forward-Backward

Foo

1.3 Predicate Logic

1.3.1 Predicates

Foo

1.3.2 Quantifiers

Foo

1.3.3 Translation

Foo

1.3.4 Negation

Foo

1.3.5 Two Common Constructions

Foo

1.4 Logic in Mathematics

1.4.1 The Role of Definitions in Mathematics

Foo

1.4.2 Other Types of Mathematical Statements

Foo

1.4.3 Counterexamples

Foo

1.4.4 Axiomatic System

Foo

1.5 Methods of Proof

1.5.1 Direct Proofs

Foo

1.5.2 Proof by Contraposition

Foo

1.5.3 Proof by Contradiction

Foo

Chapter 2

Relational Thinking

2.1 Graph

2.1.1 Edges and Vertices

Foo

2.1.2 Terminology

Foo

2.1.3 Modeling Relationships with Graphs

Foo

2.2 Sets

2.2.1 Membership and Containment

Foo

2.2.2 New Sets from Old

Foo

2.2.3 Identities

Foo

2.3 Functions

2.3.1 Definition and Examples

Foo

2.3.2 One-to-One and Onto Functions

Foo

2.3.3 New Functions from Old

Foo

2.4 Relations and Equivalences

2.4.1 Definition and Examples

Foo

2.4.2 Graphs of Relations

Foo

2.4.3 Relations vs. Functions

Foo

2.4.4 Equivalence Relations

Foo

2.4.5 Modular Arithmetic

Foo

2.5 Partial Orderings

2.5.1 Definition and Examples

Foo

2.5.2 Hasse Diagrams

Foo

2.5.3 Topological Sorting

Foo

2.5.4 Isomorphisms

Foo

2.5.5 Boolean Algebras

Foo

2.6 Graph Theory

2.6.1 Graphs: Formal Definitions

Foo

2.6.2 Isomorphisms of Graphs

Foo

2.6.3 Degree Counting

Foo

2.6.4 Euler Paths and Circuits

Foo

2.6.5 Hamilton Paths and Circuits

Foo

2.6.6 Trees

Foo

Chapter 3

Recursive Thinking

3.1 Recurrence Relations

3.1.1 Definition and Examples

Foo

3.1.2 The Fibonacci Sequence

Foo

3.1.3 Modeling with Recurrence Relations

Foo

3.2 Closed-Form Solutions and Induction

3.2.1 Guessing a Closed-Form Solution

Foo

3.2.2 Polynomial Sequences: Using Differences

Foo

3.2.3 Inductively Verifying a Solution

Foo

3.3 Recursive Definitions

3.3.1 Definition and Examples

Foo

3.3.2 Writing Recursive Definitions

Foo

3.3.3 Recursive Geometry

Foo

3.3.4 Recursive Jokes

Foo

3.4 Proof by Induction

3.4.1 The Principle of Induction

Foo

3.4.2 Examples

Foo

3.4.3 Strong Induction

Foo

3.4.4 Structural Induction

Foo

3.5 Recursives Data Structures

3.5.1 Lists

Foo

3.5.2 Efficiency

Foo

3.5.3 Binary Search Trees Revisited

Foo

Chapter 4

Quantitative Thinking

4.1 Basic Counting Techniques

4.1.1 Addition

Foo

4.1.2 Multiplication

Foo

4.1.3 Mixing Addition and Multiplication

Foo

4.2 Selections and Arrangements

4.2.1 Permutations: The Arrangement Principle

Foo

4.2.2 Combinations: The Selection Principle

Foo

4.2.3 The Binomial Theorem

Foo

4.3 Counting with Functions

4.3.1 One-to-One Correspondences

Foo

4.3.2 The Pigeonhole Principle

Foo

4.3.3 The Generalized Pigeonhole Principle

Foo

4.3.4 Ramsey Theory

Foo

4.4 Discrete Probability

4.4.1 Definitions and Examples

Foo

4.4.2 Applications

Foo

4.4.3 Expected Value

Foo

4.5 Counting Operations in Algorithms

4.5.1 Algorithms

Foo

4.5.2 Pseudocode

Foo

4.5.3 Sequences of Operations

Foo

4.5.4 Loops

Foo

4.5.5 Arrays

Foo

4.5.6 Sorting

Foo

4.6 Estimation

4.6.1 Growth of Functions

Foo

4.6.2 Estimation Targets

Foo

4.6.3 Properties of Big- Θ

Foo

Chapter 5

Analytical Thinking

5.1 Algorithms

5.1.1 More Pseudocode

Foo

5.1.2 Preconditions and Postconditions

Foo

5.1.3 Iterative Algorithms

Foo

5.1.4 Functions and Recursive Algorithms

Foo

5.2 Three Common Types of Algorithms

5.2.1 Traversal Algorithms

Foo

5.2.2 Greedy Algorithms

Foo

5.2.3 Divide-and-Conquer Algorithms

Foo

5.3 Algorithm Complexity

5.3.1 The Good, the Bad, and the Average

Foo

5.3.2 Approximate Complexity Calculations

Foo

5.4 Bounds on Complexity

5.4.1 Algorithms as Decisions

Foo

5.4.2 A Lower Bound

Foo

5.4.3 Searching an Array

Foo

5.4.4 Searching

Foo

5.4.5 P vs. NP

Foo

5.5 Program Verification

5.5.1 Verification vs. Testing

Foo

5.5.2 Verifying Recursive Algorithms

Foo

5.5.3 Searching and Sorting

Foo

5.5.4 Towers of Hanoi

Foo

5.6 Loop Invariants

5.6.1 Verifying Iterative Algorithms

Foo

5.6.2 Searching and Sorting

Foo

5.6.3 Using Invariants to Design Algorithms

Foo

Chapter 6

Thinking through Applications

6.1 Patterns in DNA

6.1.1 Mutations and Phylogenetic Distance

Foo

6.1.2 Phylogenetic Trees

Foo

6.1.3 UPGMA

Foo

6.2 Social Networks

6.2.1 Definitions and Terminology

Foo

6.2.2 Notions of Equivalence

Foo

6.2.3 Hierarchical Clustering

Foo

6.2.4 Signed Graphs and Balance

Foo

6.3 Structure of Languages

6.3.1 Terminology

Foo

6.3.2 Finite-State Machines

Foo

6.3.3 Recursion

Foo

6.3.4 Further Issues in Linguistics

Foo

6.4 Discrete-Time Population Models

6.4.1 Recursive Models for Population Growth

Foo

6.4.2 Fixed Points, Equilibrium, and Chos

Foo

6.4.3 Predator-Prey Systems

Foo

6.4.4 The SIR Model

Foo

6.5 Twelve-Tone Music

6.5.1 Twelve-Tone Composition

Foo

6.5.2 Listing All Permutations

Foo

6.5.3 Transformations of Tone Rows

Foo

6.5.4 Equivalence Classes and Symmetry

Foo