

Notes on *Essentials*  
*of Discrete Mathematics*

Michael T. Wescoat

January 10, 2016



# Contents

Preface . . . . .	ix
<b>1 Logical Thinking</b>	<b>1</b>
1.1 Formal Logic . . . . .	1
1.1.1 Inquiry Problems . . . . .	1
1.1.2 Connectives and Propositions . . . . .	1
1.1.3 Truth Tables . . . . .	1
1.1.4 Logical Equivalences . . . . .	2
1.2 Propositional Logic . . . . .	2
1.2.1 Tautologies and Contradictions . . . . .	2
1.2.2 Derivation Rules . . . . .	2
1.2.3 Proof Sequences . . . . .	2
1.2.4 Forward-Backward . . . . .	3
1.3 Predicate Logic . . . . .	3
1.3.1 Predicates . . . . .	3
1.3.2 Quantifiers . . . . .	3
1.3.3 Translation . . . . .	3
1.3.4 Negation . . . . .	3
1.3.5 Two Common Constructions . . . . .	3
1.4 Logic in Mathematics . . . . .	3
1.4.1 The Role of Definitions in Mathematics . . . . .	3
1.4.2 Other Types of Mathematical Statements . . . . .	3
1.4.3 Counterexamples . . . . .	4
1.4.4 Axiomatic System . . . . .	4
1.5 Methods of Proof . . . . .	4
1.5.1 Direct Proofs . . . . .	4
1.5.2 Proof by Contraposition . . . . .	4
1.5.3 Proof by Contradiction . . . . .	4

<b>2</b>	<b>Relational Thinking</b>	<b>5</b>
2.1	Graph . . . . .	5
2.1.1	Edges and Vertices . . . . .	5
2.1.2	Terminology . . . . .	5
2.1.3	Modeling Relationships with Graphs . . . . .	5
2.2	Sets . . . . .	5
2.2.1	Membership and Containment . . . . .	5
2.2.2	New Sets from Old . . . . .	5
2.2.3	Identities . . . . .	6
2.3	Functions . . . . .	6
2.3.1	Definition and Examples . . . . .	6
2.3.2	One-to-One and Onto Functions . . . . .	6
2.3.3	New Functions from Old . . . . .	6
2.4	Relations and Equivalences . . . . .	6
2.4.1	Definition and Examples . . . . .	6
2.4.2	Graphs of Relations . . . . .	6
2.4.3	Relations vs. Functions . . . . .	6
2.4.4	Equivalence Relations . . . . .	6
2.4.5	Modular Arithmetic . . . . .	7
2.5	Partial Orderings . . . . .	7
2.5.1	Definition and Examples . . . . .	7
2.5.2	Hasse Diagrams . . . . .	7
2.5.3	Topological Sorting . . . . .	7
2.5.4	Isomorphisms . . . . .	7
2.5.5	Boolean Algebras . . . . .	7
2.6	Graph Theory . . . . .	7
2.6.1	Graphs: Formal Definitions . . . . .	7
2.6.2	Isomorphisms of Graphs . . . . .	7
2.6.3	Degree Counting . . . . .	8
2.6.4	Euler Paths and Circuits . . . . .	8
2.6.5	Hamilton Paths and Circuits . . . . .	8
2.6.6	Trees . . . . .	8
<b>3</b>	<b>Recursive Thinking</b>	<b>9</b>
3.1	Recurrence Relations . . . . .	9
3.1.1	Definition and Examples . . . . .	9
3.1.2	The Fibonacci Sequence . . . . .	9
3.1.3	Modeling with Recurrence Relations . . . . .	9

3.2	Closed-Form Solutions and Induction . . . . .	9
3.2.1	Guessing a Closed-Form Solution . . . . .	9
3.2.2	Polynomial Sequences: Using Differences . . . . .	9
3.2.3	Inductively Verifying a Solution . . . . .	10
3.3	Recursive Definitions . . . . .	10
3.3.1	Definition and Examples . . . . .	10
3.3.2	Writing Recursive Definitions . . . . .	10
3.3.3	Recursive Geometry . . . . .	10
3.3.4	Recursive Jokes . . . . .	10
3.4	Proof by Induction . . . . .	10
3.4.1	The Principle of Induction . . . . .	10
3.4.2	Examples . . . . .	10
3.4.3	Strong Induction . . . . .	10
3.4.4	Structural Induction . . . . .	11
3.5	Recursive Data Structures . . . . .	11
3.5.1	Lists . . . . .	11
3.5.2	Efficiency . . . . .	11
3.5.3	Binary Search Trees Revisited . . . . .	11
<b>4</b>	<b>Quantitative Thinking</b>	<b>13</b>
4.1	Basic Counting Techniques . . . . .	13
4.1.1	Addition . . . . .	13
4.1.2	Multiplication . . . . .	13
4.1.3	Mixing Addition and Multiplication . . . . .	13
4.2	Selections and Arrangements . . . . .	13
4.2.1	Permutations: The Arrangement Principle . . . . .	13
4.2.2	Combinations: The Selection Principle . . . . .	13
4.2.3	The Binomial Theorem . . . . .	14
4.3	Counting with Functions . . . . .	14
4.3.1	One-to-One Correspondences . . . . .	14
4.3.2	The Pigeonhole Principle . . . . .	14
4.3.3	The Generalized Pigeonhole Principle . . . . .	14
4.3.4	Ramsey Theory . . . . .	14
4.4	Discrete Probability . . . . .	14
4.4.1	Definitions and Examples . . . . .	14
4.4.2	Applications . . . . .	14
4.4.3	Expected Value . . . . .	14
4.5	Counting Operations in Algorithms . . . . .	15

4.5.1	Algorithms . . . . .	15
4.5.2	Pseudocode . . . . .	15
4.5.3	Sequences of Operations . . . . .	15
4.5.4	Loops . . . . .	15
4.5.5	Arrays . . . . .	15
4.5.6	Sorting . . . . .	15
4.6	Estimation . . . . .	15
4.6.1	Growth of Functions . . . . .	15
4.6.2	Estimation Targets . . . . .	15
4.6.3	Properties of Big- $\Theta$ . . . . .	15
<b>5</b>	<b>Analytical Thinking</b>	<b>17</b>
5.1	Algorithms . . . . .	17
5.1.1	More Pseudocode . . . . .	17
5.1.2	Preconditions and Postconditions . . . . .	17
5.1.3	Iterative Algorithms . . . . .	17
5.1.4	Functions and Recursive Algorithms . . . . .	17
5.2	Three Common Types of Algorithms . . . . .	17
5.2.1	Traversal Algorithms . . . . .	17
5.2.2	Greedy Algorithms . . . . .	18
5.2.3	Divide-and-Conquer Algorithms . . . . .	18
5.3	Algorithm Complexity . . . . .	18
5.3.1	The Good, the Bad, and the Average . . . . .	18
5.3.2	Approximate Complexity Calculations . . . . .	18
5.4	Bounds on Complexity . . . . .	18
5.4.1	Algorithms as Decisions . . . . .	18
5.4.2	A Lower Bound . . . . .	18
5.4.3	Searching an Array . . . . .	18
5.4.4	Searching . . . . .	18
5.4.5	$P$ vs. $NP$ . . . . .	19
5.5	Program Verification . . . . .	19
5.5.1	Verification vs. Testing . . . . .	19
5.5.2	Verifying Recursive Algorithms . . . . .	19
5.5.3	Searching and Sorting . . . . .	19
5.5.4	Towers of Hanoi . . . . .	19
5.6	Loop Invariants . . . . .	19
5.6.1	Verifying Iterative Algorithms . . . . .	19
5.6.2	Searching and Sorting . . . . .	19

5.6.3	Using Invariants to Design Algorithms . . . . .	19
<b>6</b>	<b>Thinking through Applications</b>	<b>21</b>
6.1	Patterns in DNA . . . . .	21
6.1.1	Mutations and Phylogenetic Distance . . . . .	21
6.1.2	Phylogenetic Trees . . . . .	21
6.1.3	UPGMA . . . . .	21
6.2	Social Networks . . . . .	21
6.2.1	Definitions and Terminology . . . . .	21
6.2.2	Notions of Equivalence . . . . .	21
6.2.3	Hierarchical Clustering . . . . .	22
6.2.4	Signed Graphs and Balance . . . . .	22
6.3	Structure of Languages . . . . .	22
6.3.1	Terminology . . . . .	22
6.3.2	Finite-State Machines . . . . .	22
6.3.3	Recursion . . . . .	22
6.3.4	Further Issues in Linguistics . . . . .	22
6.4	Discrete-Time Population Models . . . . .	22
6.4.1	Recursive Models for Population Growth . . . . .	22
6.4.2	Fixed Points, Equilibrium, and Chos . . . . .	22
6.4.3	Predator-Prey Systems . . . . .	23
6.4.4	The SIR Model . . . . .	23
6.5	Twelve-Tone Music . . . . .	23
6.5.1	Twelve-Tone Composition . . . . .	23
6.5.2	Listing All Permutations . . . . .	23
6.5.3	Transformations of Tone Rows . . . . .	23
6.5.4	Equivalence Classes and Symmetry . . . . .	23





# Caution

The present document is not intended for publication. It is a collection of study notes on *Essentials of Discrete Mathematics*, 3rd ed., by David J. Hunter. These notes freely quote from the book without proper citation. Consequently, if anyone plucks any part out of this document for inclusion in another work, there is a high probability that she will find herself plagiarizing Hunter's original text.



# Chapter 1

## Logical Thinking

### 1.1 Formal Logic

*Formalism* is the practice of reducing a problem to a notation with well-defined symbolic manipulations.

#### 1.1.1 Inquiry Problems

...

#### 1.1.2 Connectives and Propositions

**Definition 1.** A statement (*also known as a proposition*) is a declarative sentence that is either true or false, but not both.

A complicated statement may consist of several simple statements joined together by words such as *and*, *or*, *if...then*, etc. These connecting words are represented by the five *logical connectives* shown in Table 1.1.

#### 1.1.3 Truth Tables

We must define the meaning of the logical connectives.

- Any statement has two possible values: true (T) and false (F).
- We can use variables such as  $p$  and  $q$  for statements.

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

Figure 1.1: The Five Logical Connectives

- We can stipulate the meaning of each logical connective with *truth tables* listing T/F values for every possible case.

$p$	$\neg p$
T	F
F	T

$p$	$q$	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

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

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

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

### 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- $\Theta$

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