

# Introduction to the Theory of Computation

Michael Sipser

Now Available from PWS Publishing Company.

Errata

---

## CONTENTS OF THE FIRST EDITION

---

### 0. Introduction

1. AUTOMATA, COMPUTABILITY, AND COMPLEXITY  
*Complexity theory - Computability theory - Automata theory*
2. MATHEMATICAL NOTIONS AND TERMINOLOGY  
*Sets - Sequences and tuples - Functions and relations - Graphs - Strings and languages - Boolean logic - Summary of mathematical terms*
3. DEFINITIONS, THEOREMS, AND PROOFS  
*Finding proofs*
4. TYPES OF PROOF  
*Proof by construction - Proof by contradiction - Proof by induction*

### PART ONE: AUTOMATA AND LANGUAGES

#### 1. Regular Languages

1. FINITE AUTOMATA  
*Formal definition of a finite automaton - Examples of finite automata - Formal definition of computation - Designing finite automata - The regular operations*
2. NONDETERMINISM  
*Equivalence of NFAs and DFAs - Closure under the regular operations*
3. REGULAR EXPRESSIONS  
*Formal definition of a regular expression - Equivalence with finite automata*
4. NONREGULAR LANGUAGES  
*The pumping lemma for regular languages*

#### 2. Context-Free Languages

1. CONTEXT-FREE GRAMMARS  
*Formal definition of a context-free grammar - Examples of context-free grammars - Designing context-free grammars - Ambiguity - Chomsky normal form*
2. PUSHDOWN AUTOMATA  
*Formal definition of a pushdown automaton - Examples of pushdown automata - Equivalence with context-free grammars*
3. NON-CONTEXT-FREE LANGUAGES  
*The pumping lemma for context-free languages*

## **PART TWO: COMPUTABILITY THEORY**

### **3. The Church-Turing Thesis**

1. TURING MACHINES  
*Formal definition of a Turing machine - Examples of Turing machines*
2. VARIANTS OF TURING MACHINES  
*Multitape Turing machines - Nondeterministic Turing machines - Enumerators - Equivalence with other models*
3. THE DEFINITION OF ALGORITHM  
*Hilbert's problems - Terminology for describing Turing machines*

### **4. Decidability**

1. DECIDABLE LANGUAGES
2. THE HALTING PROBLEM  
*The diagonalization method - The halting problem is undecidable - A Turing-unacceptable language*

### **5. Reducibility**

1. UNDECIDABLE PROBLEMS FROM  $\forall$  LANGUAGE THEORY  
*Reductions via computation histories*
2. A SIMPLE UNDECIDABLE PROBLEM
3. MAPPING REDUCIBILITY  
*Computable functions - Formal definition of mapping reducibility*

### **6. Advanced Topics in Computability Theory**

1. THE RECURSION THEOREM  
*Self-reference - Terminology for the recursion theorem - Applications*
2. DECIDABILITY OF LOGICAL THEORIES  
*A decidable theory - An undecidable theory*
3. TURING REDUCIBILITY
4. A DEFINITION OF INFORMATION  
*Minimal length descriptions - Optimality of the definition - Incompressible strings and randomness*

## **PART THREE: COMPLEXITY THEORY**

### **7. Time Complexity**

1. MEASURING COMPLEXITY  
*Big-O and small-o notation - Analyzing algorithms - Complexity relationships among models*
2. THE CLASS P  
*Polynomial time - Examples of problems in P*
3. THE CLASS NP  
*Examples of problems in NP - The P versus NP question*
4. NP-COMPLETENESS  
*Polynomial time reducibility - Definition of NP-completeness - The Cook-Levin Theorem*
5. EXAMPLES OF NP-COMplete PROBLEMS  
*The vertex cover problem - The Hamiltonian path problem - The subset sum problem*

### **8. Space Complexity**

1. SAVITCH'S THEOREM
2. THE CLASS PSPACE
3. PSPACE-COMPLETENESS  
*The TQBF problem - Winning strategies for games - Generalized geography*
4. THE CLASSES L AND NL
5. NL-COMPLETENESS  
*Searching in graphs*
6. NL EQUALS CONL

### **9. Intractability**

1. HIERARCHY THEOREMS  
*Exponential space completeness*
2. RELATIVIZATION  
*Limits of the diagonalization method*
3. CIRCUIT COMPLEXITY

### **10. Advanced Topics in Complexity Theory**

1. APPROXIMATION ALGORITHMS
2. PROBABILISTIC ALGORITHMS  
*The class BPP - Primality - Read-once branching programs*
3. ALTERNATION  
*Alternating time and space - The Polynomial time hierarchy*
4. INTERACTIVE PROOF SYSTEMS  
*Graph nonisomorphism - Definition of the model -  $IP = PSPACE$*

5. PARALLEL COMPUTATION

*Uniform Boolean circuits - The class NC - P-completeness*

6. CRYPTOGRAPHY

*Secret keys - Public-key cryptosystems - One-way functions - Trapdoor functions*