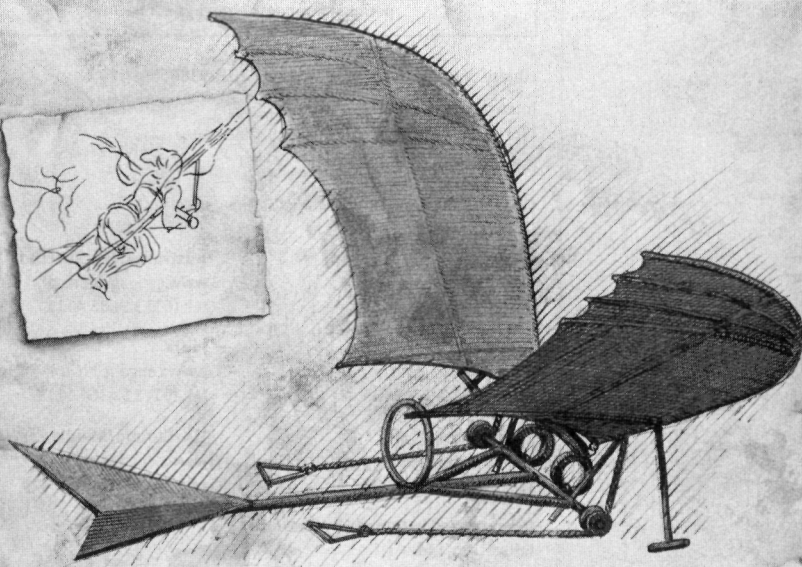


# Introduction to the Theory of COMPUTATION

THIRD EDITION



MICHAEL SIPSER



CENGAGE  
Learning®

---

Australia • Brazil • Japan • Korea • Mexico • Singapore • Spain • United Kingdom • United States

# C O N T E N T S

|                                     |           |
|-------------------------------------|-----------|
| <b>Preface to the First Edition</b> | <b>xi</b> |
| To the student . . . . .            | xi        |
| To the educator . . . . .           | xii       |
| The first edition . . . . .         | xiii      |
| Feedback to the author . . . . .    | xiii      |
| Acknowledgments . . . . .           | xiv       |

|                                      |             |
|--------------------------------------|-------------|
| <b>Preface to the Second Edition</b> | <b>xvii</b> |
|--------------------------------------|-------------|

|                                     |            |
|-------------------------------------|------------|
| <b>Preface to the Third Edition</b> | <b>xxi</b> |
|-------------------------------------|------------|

|   |          |
|---|----------|
| <b>0 Introduction</b>                                 | <b>1</b> |
| 0.1 Automata, Computability, and Complexity . . . . . | 1        |
| Complexity theory . . . . .                           | 2        |
| Computability theory . . . . .                        | 3        |
| Automata theory . . . . .                             | 3        |
| 0.2 Mathematical Notions and Terminology . . . . .    | 3        |
| Sets . . . . .  | 3        |
| Sequences and tuples . . . . .                        | 6        |
| Functions and relations . . . . .                     | 7        |
| Graphs . . . . .                                      | 10       |
| Strings and languages . . . . .                       | 13       |
| Boolean logic . . . . .                               | 14       |
| Summary of mathematical terms . . . . .               | 16       |
| 0.3 Definitions, Theorems, and Proofs . . . . .       | 17       |
| Finding proofs . . . . .                              | 17       |
| 0.4 Types of Proof . . . . .                          | 21       |
| Proof by construction . . . . .                       | 21       |
| Proof by contradiction . . . . .                      | 21       |
| Proof by induction . . . . .                          | 22       |
| <i>Exercises, Problems, and Solutions</i> . . . . .   | 25       |

|  |            |
|--|------------|
| <b>Part One: Automata and Languages</b>                            | <b>29</b>  |
| <b>1 Regular Languages</b>   | <b>31</b>  |
| 1.1 Finite Automata . . . . .                                      | 31         |
| Formal definition of a finite automaton . . . . .                  | 35         |
| Examples of finite automata . . . . .                              | 37         |
| Formal definition of computation . . . . .                         | 40         |
| Designing finite automata . . . . .                                | 41         |
| The regular operations . . . . .                                   | 44         |
| 1.2 Nondeterminism . . . . .                                       | 47         |
| Formal definition of a nondeterministic finite automaton . . . . . | 53         |
| Equivalence of NFAs and DFAs . . . . .                             | 54         |
| Closure under the regular operations . . . . .                     | 58         |
| 1.3 Regular Expressions . . . . .                                  | 63         |
| Formal definition of a regular expression . . . . .                | 64         |
| Equivalence with finite automata . . . . .                         | 66         |
| 1.4 Nonregular Languages . . . . .                                 | 77         |
| The pumping lemma for regular languages . . . . .                  | 77         |
| <i>Exercises, Problems, and Solutions</i> . . . . .                | 82         |
| <b>2 Context-Free Languages</b>                                    | <b>101</b> |
| 2.1 Context-Free Grammars . . . . .                                | 102        |
| Formal definition of a context-free grammar . . . . .              | 104        |
| Examples of context-free grammars . . . . .                        | 105        |
| Designing context-free grammars . . . . .                          | 106        |
| Ambiguity . . . . .  | 107        |
| Chomsky normal form . . . . .                                      | 108        |
| 2.2 Pushdown Automata . . . . .                                    | 111        |
| Formal definition of a pushdown automaton . . . . .                | 113        |
| Examples of pushdown automata . . . . .                            | 114        |
| Equivalence with context-free grammars . . . . .                   | 117        |
| 2.3 Non-Context-Free Languages . . . . .                           | 125        |
| The pumping lemma for context-free languages . . . . .             | 125        |
| 2.4 Deterministic Context-Free Languages . . . . .                 | 130        |
| Properties of DCFLs . . . . .                                      | 133        |
| Deterministic context-free grammars . . . . .                      | 135        |
| Relationship of DPDAs and DCFGs . . . . .                          | 146        |
| Parsing and LR(k) Grammars . . . . .                               | 151        |
| <i>Exercises, Problems, and Solutions</i> . . . . .                | 154        |
| <b>Part Two: Computability Theory</b>                              | <b>163</b> |
| <b>3 The Church-Turing Thesis</b>                                  | <b>165</b> |
| 3.1 Turing Machines . . . . .                                      | 165        |
| Formal definition of a Turing machine . . . . .                    | 167        |

|  |            |
|--|------------|
| Examples of Turing machines . . . . .                          | 170        |
| 3.2 Variants of Turing Machines . . . . .                      | 176        |
| Multitape Turing machines . . . . .                            | 176        |
| Nondeterministic Turing machines . . . . .                     | 178        |
| Enumerators . . . . .  | 180        |
| Equivalence with other models . . . . .                        | 181        |
| 3.3 The Definition of Algorithm . . . . .                      | 182        |
| Hilbert's problems . . . . .                                   | 182        |
| Terminology for describing Turing machines . . . . .           | 184        |
| <i>Exercises, Problems, and Solutions</i> . . . . .            | 187        |
| <b>4 Decidability</b>  | <b>193</b> |
| 4.1 Decidable Languages . . . . .                              | 194        |
| Decidable problems concerning regular languages . . . . .      | 194        |
| Decidable problems concerning context-free languages . . . . . | 198        |
| 4.2 Undecidability . . . . .                                   | 201        |
| The diagonalization method . . . . .                           | 202        |
| An undecidable language . . . . .                              | 207        |
| A Turing-unrecognizable language . . . . .                     | 209        |
| <i>Exercises, Problems, and Solutions</i> . . . . .            | 210        |
| <b>5 Reducibility</b>  | <b>215</b> |
| 5.1 Undecidable Problems from Language Theory . . . . .        | 216        |
| Reductions via computation histories . . . . .                 | 220        |
| 5.2 A Simple Undecidable Problem . . . . .                     | 227        |
| 5.3 Mapping Reducibility . . . . .                             | 234        |
| Computable functions . . . . .                                 | 234        |
| Formal definition of mapping reducibility . . . . .            | 235        |
| <i>Exercises, Problems, and Solutions</i> . . . . .            | 239        |
| <b>6 Advanced Topics in Computability Theory</b>               | <b>245</b> |
| 6.1 The Recursion Theorem . . . . .                            | 245        |
| Self-reference . . . . .                                       | 246        |
| Terminology for the recursion theorem . . . . .                | 249        |
| Applications . . . . .   | 250        |
| 6.2 Decidability of logical theories . . . . .                 | 252        |
| A decidable theory . . . . .                                   | 255        |
| An undecidable theory . . . . .                                | 257        |
| 6.3 Turing Reducibility . . . . .                              | 260        |
| 6.4 A Definition of Information . . . . .                      | 261        |
| Minimal length descriptions . . . . .                          | 262        |
| Optimality of the definition . . . . .                         | 266        |
| Incompressible strings and randomness . . . . .                | 267        |
| <i>Exercises, Problems, and Solutions</i> . . . . .            | 270        |

|   |            |
|---|------------|
| <b>Part Three: Complexity Theory</b>                | <b>273</b> |
| <b>7 Time Complexity</b>                            | <b>275</b> |
| 7.1 Measuring Complexity . . . . .                  | 275        |
| Big- $O$ and small- $o$ notation . . . . .          | 276        |
| Analyzing algorithms . . . . .                      | 279        |
| Complexity relationships among models . . . . .     | 282        |
| 7.2 The Class $P$ . . . . .                         | 284        |
| Polynomial time . . . . .                           | 284        |
| Examples of problems in $P$ . . . . .               | 286        |
| 7.3 The Class $NP$ . . . . .                        | 292        |
| Examples of problems in $NP$ . . . . .              | 295        |
| The $P$ versus $NP$ question . . . . .              | 297        |
| 7.4 $NP$ -completeness . . . . .                    | 299        |
| Polynomial time reducibility . . . . .              | 300        |
| Definition of $NP$ -completeness . . . . .          | 304        |
| The Cook-Levin Theorem . . . . .                    | 304        |
| 7.5 Additional $NP$ -complete Problems . . . . .    | 311        |
| The vertex cover problem . . . . .                  | 312        |
| The Hamiltonian path problem . . . . .              | 314        |
| The subset sum problem . . . . .                    | 319        |
| <i>Exercises, Problems, and Solutions</i> . . . . . | 322        |
| <b>8 Space Complexity</b>                           | <b>331</b> |
| 8.1 Savitch's Theorem . . . . .                     | 333        |
| 8.2 The Class $PSPACE$ . . . . .                    | 336        |
| 8.3 $PSPACE$ -completeness . . . . .                | 337        |
| The TQBF problem . . . . .                          | 338        |
| Winning strategies for games . . . . .              | 341        |
| Generalized geography . . . . .                     | 343        |
| 8.4 The Classes $L$ and $NL$ . . . . .              | 348        |
| 8.5 $NL$ -completeness . . . . .                    | 351        |
| Searching in graphs . . . . .                       | 353        |
| 8.6 $NL$ equals $coNL$ . . . . .                    | 354        |
| <i>Exercises, Problems, and Solutions</i> . . . . . | 356        |
| <b>9 Intractability</b>                             | <b>363</b> |
| 9.1 Hierarchy Theorems . . . . .                    | 364        |
| Exponential space completeness . . . . .            | 371        |
| 9.2 Relativization . . . . .                        | 376        |
| Limits of the diagonalization method . . . . .      | 377        |
| 9.3 Circuit Complexity . . . . .                    | 379        |
| <i>Exercises, Problems, and Solutions</i> . . . . . | 388        |
| <b>10 Advanced Topics in Complexity Theory</b>      | <b>393</b> |
| 10.1 Approximation Algorithms . . . . .             | 393        |

|   |            |
|---|------------|
| 10.2 Probabilistic Algorithms . . . . .             | 396        |
| The class BPP . . . . .                             | 396        |
| Primality . . . . .                                 | 399        |
| Read-once branching programs . . . . .              | 404        |
| 10.3 Alternation . . . . .                          | 408        |
| Alternating time and space . . . . .                | 410        |
| The Polynomial time hierarchy . . . . .             | 414        |
| 10.4 Interactive Proof Systems . . . . .            | 415        |
| Graph nonisomorphism . . . . .                      | 415        |
| Definition of the model . . . . .                   | 416        |
| IP = PSPACE . . . . .                               | 418        |
| 10.5 Parallel Computation . . . . .                 | 427        |
| Uniform Boolean circuits . . . . .                  | 428        |
| The class NC . . . . .                              | 430        |
| P-completeness . . . . .                            | 432        |
| 10.6 Cryptography . . . . .                         | 433        |
| Secret keys . . . . .                               | 433        |
| Public-key cryptosystems . . . . .                  | 435        |
| One-way functions . . . . .                         | 435        |
| Trapdoor functions . . . . .                        | 437        |
| <i>Exercises, Problems, and Solutions</i> . . . . . | 439        |
| <b>Selected Bibliography</b>                        | <b>443</b> |
| <b>Index</b>  | <b>448</b> |