Contents

List of Figures page xi

Preface xiii

Overview xvii

To the Teacher xxi

Notations and Conventions xxv

Main Definitions and Results xxvii

1 Computational Tasks and Models 1

Teaching Notes 2

1.1 Representation 3

1.2 Computational Tasks 5

1.2.1 Search Problems 5

1.2.2 Decision Problems 6

1.2.3 Promise Problems (an Advanced Comment) 8

1.3 Uniform Models (Algorithms) 8

1.3.1 Overview and General Principles 9

1.3.2 A Concrete Model: Turing Machines 11

1.3.2.1 The Actual Model 12

1.3.2.2 The Church-Turing Thesis 16

1.3.3 Uncomputable Functions 18

1.3.3.1 On the Existence of Uncomputable Functions 18

1.3.3.2 The Halting Problem 19

1.3.3.3 A Few More Undecidability Results 21

1.3.4 Universal Algorithms 22

1.3.4.1 The Existence of Universal Algorithms 23

1.3.4.2 A Detour: Kolmogorov Complexity 24

1.3.5 Time (and Space) Complexity 26

1.3.6 Oracle Machines and Turing-Reductions 29
1.3.7 Restricted Models 31
1.4 Non-Uniform Models (Circuits and Advice) 31
1.4.1 Boolean Circuits 32
1.4.1.1 The Basic Model 32
1.4.1.2 Circuit Complexity 35
1.4.2 Machines That Take Advice 36
1.4.3 Restricted Models 37
1.4.3.1 Boolean Formulae 38
1.4.3.2 Other Restricted Classes of Circuits 39
1.5 Complexity Classes 40
Exercises 41

2 The P versus NP Question 48
Teaching Notes 49
2.1 Efficient Computation 50
2.2 The Search Version: Finding versus Checking 53
2.2.1 The Class P as a Natural Class of Search Problems 54
2.2.2 The Class NP as Another Natural Class of Search Problems 56
2.2.3 The P versus NP Question in Terms of Search Problems 57
2.3 The Decision Version: Proving versus Verifying 58
2.3.1 The Class P as a Natural Class of Decision Problems 59
2.3.2 The Class NP and NP-Proof Systems 59
2.3.3 The P versus NP Question in Terms of Decision Problems 62
2.4 Equivalence of the Two Formulations 63
2.5 Technical Comments Regarding NP 65
2.6 The Traditional Definition of NP 66
2.7 In Support of P Being Different from NP 69
2.8 Philosophical Meditations 70
Exercises 71

3 Polynomial-time Reductions 74
Teaching Notes 75
3.1 The General Notion of a Reduction 75
3.1.1 The Actual Formulation 76
3.1.2 Special Cases 77
3.1.3 Terminology and a Brief Discussion 79
3.2 Reducing Optimization Problems to Search Problems 81

3.3 Self-Reducibility of Search Problems 83
3.3.1 Examples 85
3.3.2 Self-Reducibility of NP-Complete Problems 87
3.4 Digest and General Perspective 88
Exercises 89

4 NP-Completeness 96
Teaching Notes 97
4.1 Definitions 98
4.2 The Existence of NP-Complete Problems 99
Bounded Halting and Non-Halting 102
4.3 Some Natural NP-Complete Problems 103
4.3.1 Circuit and Formula Satisfiability: CSAT and SAT 104
4.3.1.1 The NP-Completeness of CSAT 105
4.3.1.2 The NP-Completeness of SAT 109
4.3.2 Combinatorics and Graph Theory 113
4.3.3 Additional Properties of the Standard Reductions 120
4.3.4 On the Negative Application of NP-Completeness 121
4.3.5 Positive Applications of NP-Completeness 122
4.4 NP Sets That Are Neither in P nor NP-Complete 126
4.5 Reflections on Complete Problems 130
Exercises 133

5 Three Relatively Advanced Topics 142
Teaching Notes 142
5.1 Promise Problems 142
5.1.1 Definitions 143
5.1.1.1 Search Problems with a Promise 143
5.1.1.2 Decision Problems with a Promise 144
5.1.1.3 Reducibility Among Promise Problems 145
5.1.2 Applications and Limitations 146
5.1.2.1 Formulating Natural Computational Problems 146
5.1.2.2 Restricting a Computational Problem 147
5.1.2.3 Non-generic Applications 147
5.1.2.4 Limitations 148
5.1.3 The Standard Convention of Avoiding Promise Problems 149
5.2 Optimal Search Algorithms for NP 151
5.3 The Class coNP and Its Intersection with NP 154
Exercises 158

Historical Notes 165

Epilogue: A Brief Overview of Complexity Theory 169

Appendix: Some Computational Problems 177

A.1 Graphs 177

A.2 Boolean Formulae 179

Bibliography 181

Index 183