

LANGUAGES AND MACHINES

8th EDITION April 2021

by Dr C. D. H. Cooper

These notes were prepared for students at Macquarie University in Australia but are freely available to anyone. However if you make use of them and are not a Macquarie University student it would be nice if you could email me at christopherdonaldcooper@gmail.com to let me know where you are from. And, if you are from outside of Australia perhaps you could send me a postcard of where you are from to pin up on my wall (Christopher Cooper, 31 Epping Avenue, EASTWOOD, NSW 2122, Australia).

© Dr C.D.H. Cooper 2021

INTRODUCTION

The unifying theme in this book is the concept of a *language* as a system of strings of characters obeying certain rules.

Strings are fundamental to computing science. Although we tend to think of computers as devices for manipulating numbers, words and pictures, it is more accurate to think of them as machines for manipulating strings of symbols which in turn represent numbers, words or pictures.

We begin by discussing a special-purpose language — the language of logic. Then we develop a language for talking about languages in general. Since languages are just sets of strings we need to develop ways of talking about sets and the associated concepts of relations and functions, using yet another special-purpose language.

At this stage we discuss proofs. Proofs are a stumbling block to many students. We will investigate the nature of proof from the syntactic point of view, that is, thinking of proofs as sequences of strings of symbols related to each other in a mechanical way. This is not the whole picture since significant proofs require insight and imagination. But routine proofs — the sort that *you* might be called upon to construct — can be generated in a routine, mechanical way from the definitions and the logical structure of the statements to be proved. As well as helping you to write sound mathematical proofs this

training will give you some insight into automated proofs of program correctness.

Then we study the connection between languages and machines that manipulate them. Finite-state machines and Turing Machines are important mathematical models of the computing process and they are used to answer a number of important theoretical questions such as:

- * How does one give a finite description of a language if it contains infinitely many possible strings?
- * Are there any calculations which are theoretically impossible for a computer to perform?

Finally we look briefly at encryption and coding. Both of these involve transforming strings of symbols.

With encryption the concern is with electronic data being intercepted by an unauthorised party. Error-correcting codes on the other hand deal with the problem of transmitted data being accidentally corrupted by random noise in the transmission channel. Both of these are important applications to the computing and communications industries. At the same time, both make real use of interesting parts of mathematics — the mathematics of integers and of polynomials.

CONTENTS

PART A: FUNDAMENTAL CONCEPTS

1. LOGIC

1.1 The Role of Logic in Mathematics	11
1.2 The Role of Logic in Computing Science	13
1.3 Propositions and Truth Functions	14
1.4 Compound Propositions	16
1.5 Tautologies	21
1.6 Translating From English	23
1.7 Laws of Logic	24
1.8 Quantifiers	26
1.9 Quantifiers in Mathematics	31
1.10 Negation Rules	32
1.11 Writing Proofs	33
1.12 Patterns of Proof	35
1.13 The Importance of Definitions	40
Exercises for Chapter 1	48
Solutions for Chapter 1	56

2. LANGUAGES

2.1 The Languages of Mathematics	72
2.2 Languages and Computing Science	73
2.3 Strings	74
2.4 Languages	76
2.5 String Substitution	82
2.6 Formal Grammars	83
2.7 Regular Expressions	87
Exercises for Chapter 2	93
Solutions for Chapter 2	96

3. SETS, FUNCTIONS AND RELATIONS

3.1 Sets in Mathematics and Computing	101
3.2 Defining a Set	102
3.3 Basic Set Functions	106
3.4 Venn Diagrams	107
3.5 Extended Set Functions	109
3.6 Relations	110
3.7 The Sum and Product of Relations	112
3.8 Equivalence Relations	113
3.9 Equivalence Classes	116
3.10 Functions	118
3.11 One-to-one and Onto Functions	120
3.12 Counting Finite Sets	122
3.13 Counting Infinite Set.....	127
Exercises for Chapter 3	136
Solutions for Chapter 3	143

PART B: FINITE-STATE MACHINES

4. INTRODUCTION TO FINITE-STATE MACHINES

4.1 Cyclops, The Simplest Possible Computer	153
4.2 Finite State Machines	158
4.3 Mealy Machines	161
4.4 Moore Machines	165
4.5 Finite State Acceptors	166
4.6 State Diagrams	168
4.7 Black Holes	171
Exercises for Chapter 4	174
Solutions for Chapter	181

5. REDUCTION OF FSAs	
5.1 Equivalent Machines	191
5.2 Removing Inaccessible States.....	194
5.3 Equivalent States.....	198
5.4 k -equivalence of States.....	202
5.5 Locating Equivalent States.....	207
5.6 Identifying States.....	212
5.7 Standard Form	213
5.8 Reduction of Mealey and Moore Machines	215
Exercises for Chapter 5	219
Solutions for Chapter 5	225
6. NON-DETERMINISTIC FSAs	
6.1 Multiple Transitions	241
6.2 Null Transitions	244
6.3 Multiple Initial States	246
6.4 Completing the Nulls	246
6.5 Removing the Nulls	250
6.6 Combining Multiple Transitions	255
6.7 A Worked Example	259
Exercises for Chapter 6	263
Solutions for Chapter 6	267
7. FSAs AND REGULAR LANGUAGES	
7.1 Regular Languages	275
7.2 The Languages 0, 1, λ and \emptyset	276
7.3 The Product of Two Languages	277
7.4 The Sum of Two Languages	280
7.5 The Kleene Star of a Language	281

7.6 Additional Examples	282
7.7 Example of a Non-Regular Language	284
7.8 Set Operations and Regular Languages	286
Exercises for Chapter 7	296
Solutions for Chapter 7	298

PART C: COMPUTABILITY

8. TURING MACHINES

8.1 The Limits of Computing	307
8.2 The Halting Problem	309
8.3 Definition of a Turing Machine	311
8.4 Old Notation	317
8.5 Unary Representation of Numbers	320
8.6 Adding Turing Machines	326
8.7 Sample Turing Machines	328
Exercises for Chapter 8	336
Solutions for Chapter 8	341

9. EXTENDED TURING MACHINES

9.1 The Basic Turing Machine and Its Extensions	351
9.2 Multiple Tracks	354
9.3 Multiple Characters	361
9.4 Multiple Heads	365
9.5 A Universal Turing Machine	375
9.6 Computable Numbers	385
Exercises for Chapter 9	391
Solutions for Chapter 9	394

10. THE BUSY BEAVER AND HALTING PROBLEMS

10.1 The Problem	399
10.2 Why it is Unsolvable	405
10.3 The Halting Problem	409
10.4 Will That Turing machine Halt?	415
10.5 The Flying Dutchman Problem	421
Exercises for Chapter 10	424
Solutions for Chapter 10	431

PART D: CODING THEORY

11. ARITHMETIC MODULO m

11.1 Days of the Week	441
11.2 The System \mathbb{Z}_7	443
11.3 The System \mathbb{Z}_m	448
11.4 Inverses in \mathbb{Z}_m	453
11.5 Powers in \mathbb{Z}_m	458
11.6 Euler's ϕ -Function.....	462
Exercises for Chapter 11	466
Solutions for Chapter 12	467

12. PUBLIC KEY CRYPTOGRAPHY

12.1 The RSA Code: How it Works	469
12.2 The RSA Code: Why it Works	474
12.3 Is it Secure?	475
12.4 Cracking the RSA Code	477
12.5 Signature Verification	479
Exercises for Chapter 12	483
Solutions for Chapter 12	487

13. POLYNOMIAL CODES

13.1 Error Detection and Error Correction	495
13.2 Polynomials	497
13.3 Complete Polynomials	503
13.4 Polynomial Codes: How they Work	504
13.5 Polynomial Codes: Why they Work	510
13.6 Analysis of Probabilities	513
Exercises for Chapter 13	515
Solutions for Chapter 13	517

APPENDICES

A: DICTIONARY OF TERMS	523
B: PATTERNS OF PROOF	533
C: SUMMARY	535