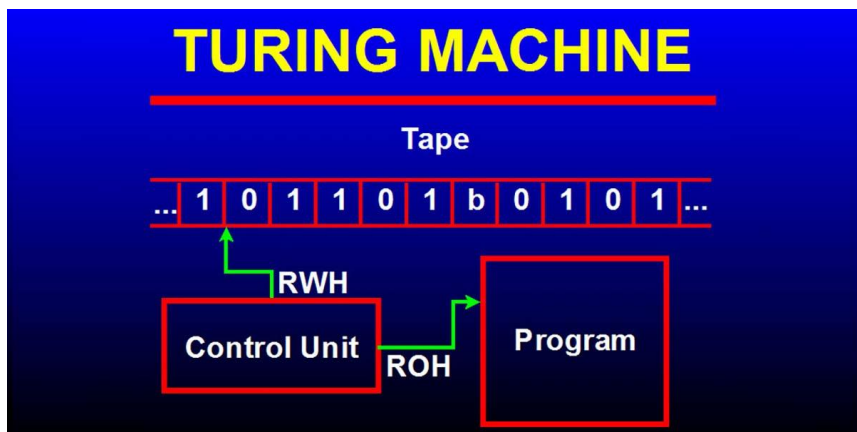


LANGUAGES AND MACHINES

TURING MACHINE



by Dr C. D. H. Cooper

9th EDITION January 2022

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 2022

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 | 12 |
| 1.3 Propositions and Truth Functions | 14 |
| 1.4 Compound Propositions | 16 |
| 1.5 Tautologies | 20 |
| 1.6 Translating From English | 22 |
| 1.7 Laws of Logic | 23 |
| 1.8 Quantifiers | 26 |
| 1.9 Quantifiers in Mathematics | 30 |
| 1.10 Negation Rules | 31 |
| 1.11 Writing Proofs | 32 |
| 1.12 Patterns of Proof | 34 |
| 1.13 The Importance of Definitions | 39 |
| Exercises for Chapter 1 | 47 |
| Solutions for Chapter 1 | 55 |

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 | 86 |
| Exercises for Chapter 2 | 92 |
| Solutions for Chapter 2 | 95 |

3. SETS, FUNCTIONS AND RELATIONS

| | |
|---|-----|
| 3.1 Sets in Mathematics and Computing | 99 |
| 3.2 Defining a Set | 100 |
| 3.3 Basic Set Functions | 104 |
| 3.4 Venn Diagrams | 105 |
| 3.5 Extended Set Functions | 107 |
| 3.6 Relations | 108 |
| 3.7 The Sum and Product of Relations | 110 |
| 3.8 Equivalence Relations | 111 |
| 3.9 Equivalence Classes | 113 |
| 3.10 Functions | 116 |
| 3.11 One-to-one and Onto Functions | 118 |
| 3.12 Counting Finite Sets | 119 |
| 3.13 Counting Infinite Set..... | 125 |
| Exercises for Chapter 3 | 134 |
| Solutions for Chapter 3 | 141 |

PART B: FINITE-STATE MACHINES

4. INTRODUCTION TO FINITE-STATE MACHINES

| | |
|---|-----|
| 4.1 Cyclops, The Simplest Possible Computer | 151 |
| 4.2 Finite State Machines | 156 |
| 4.3 Mealy Machines | 159 |
| 4.4 Moore Machines | 163 |
| 4.5 Finite State Acceptors | 164 |
| 4.6 State Diagrams | 166 |
| 4.7 Black Holes | 169 |
| Exercises for Chapter 4 | 172 |
| Solutions for Chapter | 179 |

| | |
|---|-----|
| 5. REDUCTION OF FSAs | |
| 5.1 Equivalent Machines | 189 |
| 5.2 Removing Inaccessible States..... | 192 |
| 5.3 Equivalent States..... | 196 |
| 5.4 k -equivalence of States..... | 201 |
| 5.5 Locating Equivalent States..... | 205 |
| 5.6 Identifying States..... | 210 |
| 5.7 Standard Form | 211 |
| 5.8 Reduction of Mealey and Moore Machines | 213 |
| Exercises for Chapter 5 | 217 |
| Solutions for Chapter 5 | 223 |
| | |
| 6. NON-DETERMINISTIC FSAs | |
| 6.1 Multiple Transitions | 239 |
| 6.2 Null Transitions | 242 |
| 6.3 Multiple Initial States | 244 |
| 6.4 Completing the Nulls | 244 |
| 6.5 Removing the Nulls | 248 |
| 6.6 Combining Multiple Transitions | 253 |
| 6.7 A Worked Example | 257 |
| Exercises for Chapter 6 | 261 |
| Solutions for Chapter 6 | 265 |
| | |
| 7. FSAs AND REGULAR LANGUAGES | |
| 7.1 Regular Languages | 273 |
| 7.2 The Languages 0, 1, λ and \emptyset | 274 |
| 7.3 The Product of Two Languages | 275 |
| 7.4 The Sum of Two Languages | 278 |
| 7.5 The Kleene Star of a Language | 280 |

| | |
|---|-----|
| 7.6 Additional Examples | 281 |
| 7.7 Example of a Non-Regular Language | 283 |
| 7.8 Set Operations and Regular Languages | 285 |
| Exercises for Chapter 7 | 294 |
| Solutions for Chapter 7 | 297 |

PART C: COMPUTABILITY

8. TURING MACHINES

| | |
|---|-----|
| 8.1 The Limits of Computing | 305 |
| 8.2 The Halting Problem | 307 |
| 8.3 Definition of a Turing Machine | 309 |
| 8.4 Old Notation | 315 |
| 8.5 Unary Representation of Numbers | 318 |
| 8.6 Adding Turing Machines | 324 |
| 8.7 Sample Turing Machines | 326 |
| Exercises for Chapter 8 | 333 |
| Solutions for Chapter 8 | 339 |

9. EXTENDED TURING MACHINES

| | |
|--|-----|
| 9.1 The Basic Turing Machine and Its Extensions | 349 |
| 9.2 Multiple Tracks | 352 |
| 9.3 Multiple Characters | 359 |
| 9.4 Multiple Heads | 363 |
| 9.5 A Universal Turing Machine | 373 |
| 9.6 Computable Numbers | 383 |
| Exercises for Chapter 9 | 389 |
| Solutions for Chapter 9 | 392 |

10. THE BUSY BEAVER AND HALTING PROBLEMS

| | |
|---|-----|
| 10.1 The Problem | 397 |
| 10.2 Why it is Unsolvable | 403 |
| 10.3 The Halting Problem | 407 |
| 10.4 Will That Turing machine Halt? | 413 |
| 10.5 The Flying Dutchman Problem | 419 |
| Exercises for Chapter 10 | 422 |
| Solutions for Chapter 10 | 429 |

PART D: CODING THEORY

11. ARITHMETIC MODULO m

| | |
|---------------------------------------|-----|
| 11.1 Days of the Week | 439 |
| 11.2 The System \mathbb{Z}_7 | 441 |
| 11.3 The System \mathbb{Z}_m | 445 |
| 11.4 Inverses in \mathbb{Z}_m | 450 |
| 11.5 Powers in \mathbb{Z}_m | 455 |
| 11.6 Euler's ϕ -Function..... | 459 |
| Exercises for Chapter 11 | 463 |
| Solutions for Chapter 12 | 465 |

12. PUBLIC KEY CRYPTOGRAPHY

| | |
|---------------------------------------|-----|
| 12.1 The RSA Code: How it Works | 467 |
| 12.2 The RSA Code: Why it Works | 472 |
| 12.3 Is it Secure? | 473 |
| 12.4 Cracking the RSA Code | 474 |
| 12.5 Signature Verification | 477 |
| Exercises for Chapter 12 | 481 |
| Solutions for Chapter 12 | 485 |

13. POLYNOMIAL CODES

| | |
|---|-----|
| 13.1 Error Detection and Error Correction | 493 |
| 13.2 Polynomials | 495 |
| 13.3 Complete Polynomials | 501 |
| 13.4 Polynomial Codes: How they Work | 502 |
| 13.5 Polynomial Codes: Why they Work | 508 |
| 13.6 Analysis of Probabilities | 511 |
| Exercises for Chapter 13 | 513 |
| Solutions for Chapter 13 | 515 |

APPENDICES

| | |
|-------------------------------------|-----|
| A: DICTIONARY OF TERMS | 521 |
| B: PATTERNS OF PROOF | 531 |
| C: SUMMARY | 533 |