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).

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

2. LANGUAGES

3. SETS, FUNCTIONS AND RELATIONS

PART B: FINITE-STATE MACHINES

4. INTRODUCTION TO FINITE-STATE MACHINES

5. REDUCTION OF FSAs

6. NON-DETERMINISTIC FSAs

7. FSAs AND REGULAR LANGUAGES

PART C: COMPUTABILITY

8. TURING MACHINES

9. EXTENDED TURING MACHINES

13. POLYNOMIAL CODES

APPENDICES