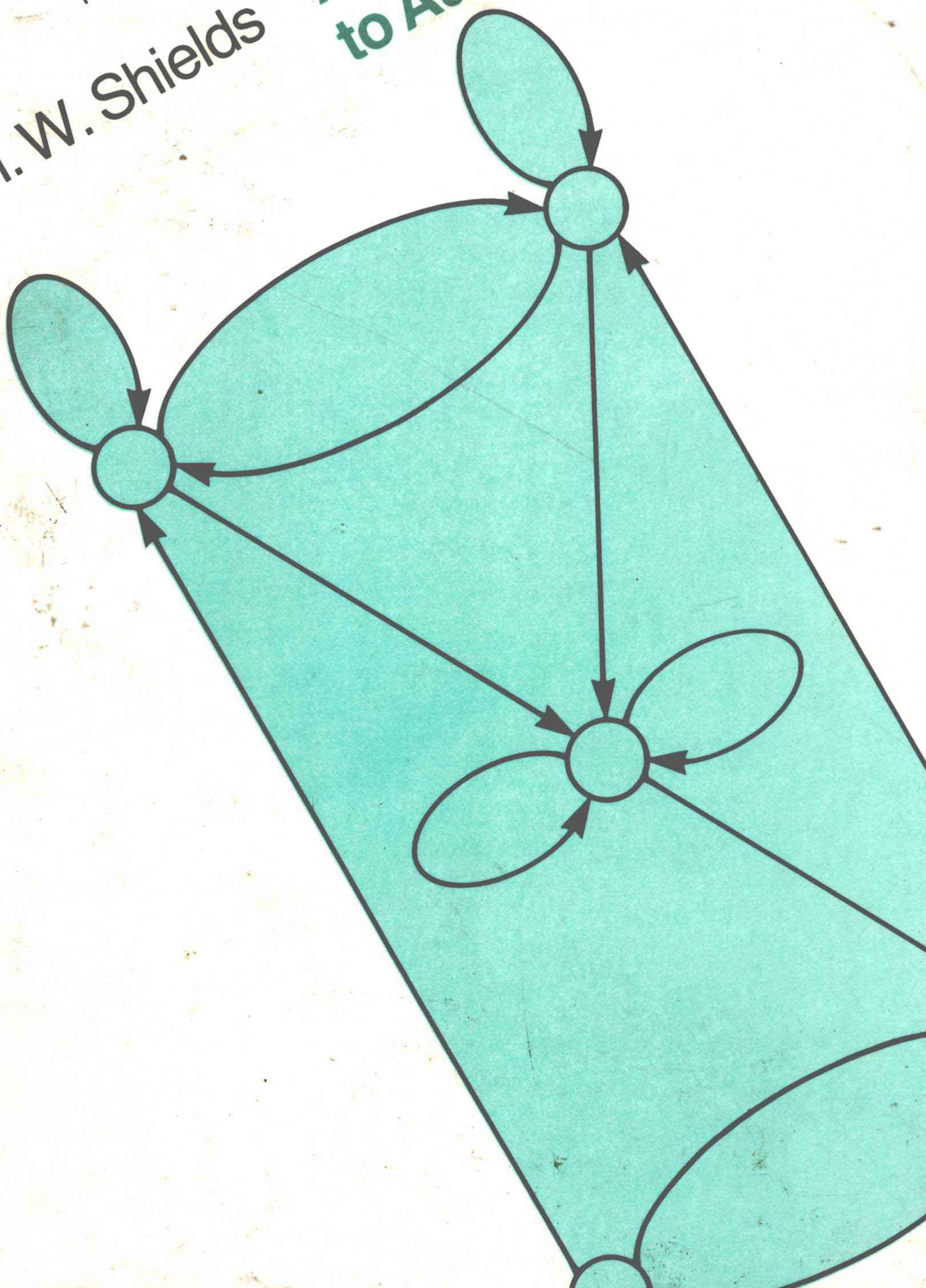
Blackwell
Scientific
Publications

Computer
Science
Texts

An Introduction to Automata Theory

M. W. Shields



COMPUTER SCIENCE TEXTS

AN INTRODUCTION TO AUTOMATA THEORY

M. W. SHIELDS
BA, PhD
Lecturer, Electronic Engineering Laboratories,
The University of Kent at Canterbury

BLACKWELL SCIENTIFIC PUBLICATIONS

OXFORD LONDON EDINBURGH

BOSTON PALO ALTO MELBOURNE

Editorial offices:
Osney Mead, Oxford OX2 0EL
(*Orders:* Tel. 0865 240201)
8 John Street, London WC1N 2ES
23 Ainslie Place, Edinburgh EH3 6AJ
52 Beacon Street, Boston
Massachusetts 02108, USA
667 Lytton Avenue, Palo Alto
California 94301, USA
107 Barry Street, Carlton
Victoria 3053, Australia

First published 1987

Printed and bound in Great Britain

DISTRIBUTORS

USA and Canada
Blackwell Scientific Publications Inc
PO Box 50009, Palo Alto
California 94303
(*Orders:* Tel. (415) 965-4081)

Australia
Blackwell Scientific Publications
(Australia) Pty Ltd
107 Barry Street
Carlton, Victoria 3053
(*Orders:* Tel. (03) 347 0300)

British Library
Cataloguing in Publication Data

Shields, M.W.
An introduction to automata theory.
1. Machine theory
I. Title
001.53'5 QA267

ISBN 0–632–01756–2
ISBN 0–632–01553–3 Pbk

Library of Congress
Cataloging-in-Publication Data

Shields, M. W. (Mike W.)
An introduction to automata theory.

Bibliography: p.
Includes index.
1. Machine theory. I. Title.
QA267.S47 1987 511.3 87–27740
ISBN 0–632–01756–2

AN INTRODUCTION TO AUTOMATA THEORY

COMPUTER SCIENCE TEXTS

To my Mother,

this little book is respectfully dedicated.

Preface

Welcome to the machine.

PINK FLOYD
Wish you were here

This book is based on a course of ten one-hour lectures given by the author to third year computer science and computer systems engineering students at the University of Kent. Ten hours is not very much, certainly not enough to present the usual material on automata theory – the Chomsky hierarchy and the related hierarchy of machines, Turing computability and so on – and so I decided to teach some elementary structure theory. My aim was to show that *systems*, of which students had some experience, were capable of mathematical manipulation of a fairly non-trivial kind, that they are not obdurate monoliths approachable only by the dubious intuition of the 'real programmer'. In other words, I attempted to present automata theory as a rudimentary *formal theory of systems*.

Thus, within its domain of application, it provides a means for *formal specification* via flow tables or state graphs and, considering these as notations for specification, automata provide a *formal semantics* for them. There are also standard techniques for *implementation* in hardware (and software). The objects with which automata are implemented may be interpreted as automata themselves and we have notions of structural relationships between automata which give us means of formalising the notion of a *correct implementation*.

It is received wisdom that non-trivial systems should not be constructed or conceived as a homogeneous whole but rather constructed out of interconnected modules. Automata theory possesses various formal notions of *composition* which enable complex systems to be constructed out of simpler ones.

There is a converse problem of *systems partitioning*: breaking a rough specification into modules which may then be refined. We devote several chapters to the problem of *decomposing* automata. It is satisfying to see to what extent we can deduce information about how a system may be decomposed from an associated lattice of congruences.

This book does not demand a great deal of mathematical sophistication from the reader, who is merely required to be comfortable with set theory and be capable of following a reasoned argument.

There is a set of exercises at the end of each chapter. Most of these are very simple and are designed to illustrate some aspect of the text. Some involve a step-by-step development of further results. There are also notes at the end of each chapter, describing extensions to the material and giving suggestions for further reading.

The author would like to thank Martin Guy for his helpful suggestions (but not for his coffee), Dan Simpson for his comments and Dominic Vaughan of Blackwell Scientific Publications for his patience. I would particularly like to thank Steve Smith for undertaking the gargantuan task of reformatting my original troff source.

Of course, any errors remaining in the text remain the responsibility of the author.

Mike Shields

Contents

Chapter 1

Great Aunt Eugenia and Other Automata

> You'll get more out of me and a machine than you will out of twenty labourers, and not so much to drink, either.
>
> GEORGE BERNARD SHAW
> *Man and Superman*

1.1 ON MACHINES

This book is about machines. The word *machine* can evoke several different pictures in peoples' minds. Mention it to a mechanical engineer, for example, and he will visualise some vast, brutish agglomerate of pulleys and cogwheels and boilers. The word has an entirely different connotation to a programmer (as in the redolent phrase, 'The machine's down again'), while to a houseperson, it would, I suppose, summon up visions of vacuum cleaners or tumble-action spin driers or (if he/she were Russian), tractors.

We shall be using the word in an *abstract* and *mathematical* sense; the word 'machine' will here denote a mathematical entity. The advantage of such an abstract approach is generality. Our results may be used in many apparently unrelated cases; what they will have in common is that in each case the same notion of machine will apply.

The more fundamental mathematical entities were usually contrived to reflect something in the real world. Whole numbers, for example, arose from the need to describe and predict the sizes of aggregates. The idea of 'machine' that we shall be using abstracts from that of a discrete, finite state, input/output device. Such a device has only a finite number of possible internal configurations. It has a specific collection of stimuli that may be applied, such as the pressing of a button or the insertion of a coin, and a finite collection of responses, for example the appearance of characters on a screen or the arrival of a packet of biscuits in a slot. Examples are: processors, programs, protocols and even (in a limited sense) people.

Such devices may be abstractly modelled by our abstract machines. But the abstract machines may themselves always be implemented by hardware and software, as we shall see. Indeed, one of the uses of abstract machines is in the design of digital hardware.

The familiar mathematical entities may usually be manipulated in some way. For example, numbers may be added or multiplied or factorized into a product of primes. Machines may also be combined to make other machines, or decomposed into simpler machines.

Why should this interest a computer scientist ?

What we are presenting here are the rudiments of a *theory of systems*. Computer science is, at the time of writing, beset by the '*software crisis*', a failure to produce software at a price, with a reliability and in a quantity sufficient to meet the needs of a voracious and increasing market. Part of the trouble is that the subject is, as yet, a craft rather than an engineering discipline. If it is ever to crawl out of the dark ages, it will need to acquire an appropriate mathematical theory of systems, as the traditional engineering disciplines have done.

Two candidates for such a theory of systems are presented in Chapters 11 and 12. They are both still subjects of extensive research.

The topics we discuss in this book reflect requirements for such a theory: the need for a notation for specification and a formal semantics for such a notation (Chapter 2); the ability to transform specification into implementation (Chapter 3); well-defined structural and behavioural relationships that permit exact comparison between systems and hence allow us to define what it means for an implementation to meet a specification (Chapters 4 and 5); the ability to compose systems together to make more complex ones (Chapter 6); the ability to partition systems to obtain a scheme of interacting modules (Chapters 7 and 8); and a mathematical theory of structure to support partitioning (Chapters 9 and 10).

Before elaborating on this – and certainly before we begin to look at precise definitions – we shall consider a couple of examples.

1.2 EXAMPLE: GREAT AUNT EUGENIA

An esteemed colleague, Dr F. X. Reid, is encumbered with a Great Aunt of advanced age, wealth and senile dementia. He reports that she responds to few stimuli; to be more precise, she reacts to three things only: ghetto-blasters* roaring out in the street outside her home, so-called video nasties† and gin‡, when a bottle of it is waved under her nose.

---

* Since I still have my sanity, it is my fervent hope that these things become obsolete and that accordingly the phrase lapses from polite use. It is in this hope that I now define the term for the sake of future generations. A ghetto-blaster is a vociferous object carried around by persons of no morals or taste to cause grief and annoyance to innocent citizens in public places.

† In a similar vein, and for a similar reason, let me explain that a video nasty is the cinematic equivalent of the ghetto-blaster.

‡ I trust that this word will not require a gloss.

These will have differing effects on her, depending on her mood. Reid tells me:

'She's a devil to get to wake up in the morning. Try anything you like – wave the gin under her nose, play "The Ghoul That Ate Guildford" – it doesn't make the slightest bit of difference. She just snores like fury and goes on sleeping, for weeks, sometimes. The family have had to buy one of those "ghetto-blasters" – that's the only thing that gets through to her.

'Mind you, she's a sweet-tempered old dear once she's woken up. Take "video nasties", for instance. She would never *dream* of doing anything as un-Victorian as watching one of the things in that mood. But if some young lout comes swaggering past the window with a "ghetto-blaster" then, my word ! She gets very irritable and starts a long harangue about bringing back ear-cropping. I'm afraid that if you offer her a bottle of gin while she's in a good mood, then that's it. She just starts swigging it and becomes embarrassingly maudlin.

'When she's in one of her irritable moods the only thing that will make her good-tempered is to watch a video nasty, though given the somewhat unpleasant cackling noise she makes, one wonders sometimes whether it's worth it. More inane cacophony doesn't help her temper or reduce her obsession with ears, but the worst of it is if she gets her hands on the gin. Glug-glug, down it goes, and there's only one word for her after that: "unspeakable".

'It doesn't matter whether she's being maudlin or unspeakable, it's still bad news for any noisy, inadvertent serenader that strolls past. She has this 1945 German army surplus luger, you see – well, I won't go on, but let's just say that I hate to think what we've had to hand over in out-of-court settlements. And she always ends up in an unspeakable mood afterwards. What we tend to do is get her to watch "The Martlesham Buzz-Saw Massacre". It doesn't sober her up, of course, and the cackling gets on one's nerves a little, but at least she's not unspeakable any more.'

'But doesn't she ever eat ?' I asked him.

'None of us have ever seen her. She seems to live on gin. I don't want you to get the wrong idea, though. Even she has a limit to the amount she can put away. Give her a second bottle and within a few minutes there she'll be, collapsed onto the bed (or the floor, whichever is nearer), dead to the world.'

All this is beginning to sound like the preamble of a 'brain-teaser' and so, to make the resemblance complete, I will add a question:

[Q1] Suppose Great Aunt Eugenia is in an unspeakable mood. How may she be rendered equable ?

As with most problems that are stated verbally (not to say verbosely), a good strategy is to present the data in a more coherent and organized form. Great Aunt Eugenia may be summed up as in Table 1.1.

Table 1.1. Great Aunt Eugenia

Moods	*Stimuli*			*Responses*		
	ghetto-b	video	gin	ghetto-b	video	gin
asleep	genial	asleep	asleep	wake	snore	snore
genial	irritable	genial	maudlin	harangue	refuse	drink
irritable	irritable	genial	unspeakable	harangue	cackle	drink
maudlin	unspeakable	maudlin	asleep	snipe	cackle	collapse
unspeakable	unspeakable	maudlin	asleep	snipe	cackle	collapse

In the table, each row corresponds to one of her five moods. The left-hand part of the table shows how a stimulus can cause her mood to change. For example, if you want to know what happens if she is offered gin when asleep, look at the intersection of the row marked 'asleep' and the column labelled 'gin'. There, you'll find the word 'asleep' – the offer has no effect. If, on the other hand, you want to know what her reaction is, look at the intersection of the 'asleep' row with the 'gin' column in the right-hand part of the table. You will see that all she does in that case is snore.

From the table it is also easy to work out the successive effects of a sequence of stimuli. Suppose, for example, that she is asleep, and that in turn we (i) play some loud rock music near to her window, (ii) offer her a bottle of gin, and (iii) repeat the rock music.

The effect of (i) will be for her to wake and become genial. In that mood, the effect of (ii) will cause her to drink and become maudlin. In the maudlin mood, she will respond to (iii) by taking pot-shots at us and her mood will become unspeakable. Thus, given the *input sequence*:

ghetto-b gin ghetto-b

and given that initially she is asleep, we will get the *output sequence*:

wake drink snipe

and her final mood will be 'unspeakable'.

[Q1] asks us to find an input sequence that will 'drive' the old lady from an unspeakable mood to one of geniality. Rather than trying to work this out from the table, it is easier to consider the same information presented pictorially, as in the diagram of Fig. 1.1.

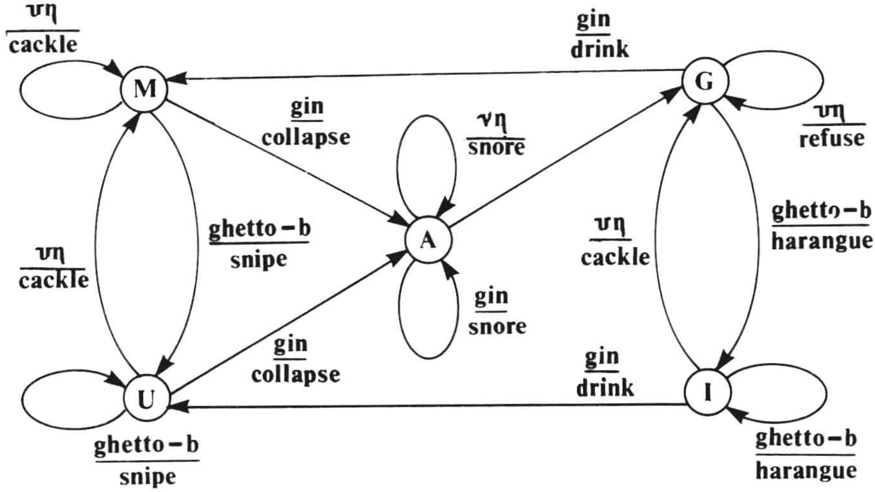


Fig. 1.1. A graphical representation of Table 1.1

The relationship between the two should be clear enough. To each mood, there is a circle, labelled by the mood. If s and s' are any two moods, and in mood s an input x will cause a change of mood to s' and produce an output y, then there is an arrow from s to s' labelled $\frac{x}{y}$.

One can see the effect of sequences of inputs by playing a *token game* on this graph. Place a counter on the initial mood. For any given input, there is exactly one arc leading from the marked mood having that input as a label. Move the token along that arc, reading off the output from the lower label on the arrow. The new position of the token will indicate her new mood.

So, to find sequences that will change her mood, we look for connected sequences of arrows that will take us, if we follow them, from the first mood to the second. If you look at the diagram, you'll see that one such sequence goes from 'unspeakable' to 'asleep' along an arrow marked $\frac{\textit{gin}}{\textit{collapse}}$ and from 'asleep' to 'genial' via an arrow marked $\frac{\textit{ghetto-b}}{\textit{wake}}$.

So one answer to question [Q1] is:

[A1] Give the old girl a bottle of gin and let her sleep it off !

1.3 EXAMPLE: PIPELINES

We can also use tables such as Table 1.1 to describe the input/output behaviour of finite storage devices. In this example, we consider a three-frame binary pipeline buffer. This can hold up to three binary digits. Its action is illustrated in Fig. 1.2.

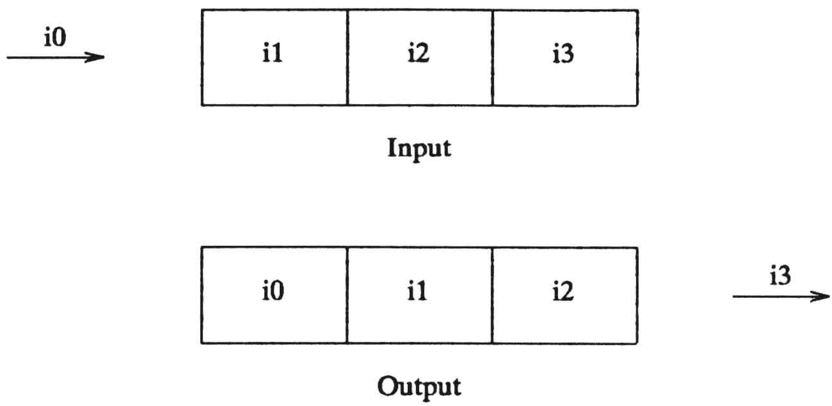


Fig. 1.2. Action of the pipeline

The action is simple enough. Binary digits are pushed in from the left and the contents of the pipeline are shifted one frame to the right. The contents of the rightmost frame emerge as output. The device acts as a three-frame shift register, eternally in shift mode.

The full definition of the input/output behaviour of the pipeline is given in Table 1.2.

Both Great Aunt Eugenia and the pipeline can be described in roughly the same way, by a table or graph describing how she or it responds to stimuli.

It should be easy to see how we might define an n-frame pipeline for any given finite number n. As an exercise, you might ask yourself how you might go about defining the general n-frame pipeline. We shall do this in Section 6.3.3.

One way of doing this that will have immediately occurred to anyone who has had even a brief acquaintance with digital hardware, is to glue several single-frame pipelines together, so that the output of each becomes the input of the next (except for the last frame, of course). We may say that the pipeline *decomposes* into three single frames or that it

Table 1.2. A three-frame pipeline

Contents	*Inputs* 0	1	*Outputs* 0	1
000	000	100	0	0
001	000	100	1	1
010	001	101	0	0
011	001	101	1	1
100	010	110	0	0
101	010	110	1	1
110	011	111	0	0
111	011	111	1	1

may be *composed* out of them. Less obviously, perhaps, Great Aunt Eugenia may also be decomposed, and because her case is less obvious, we shall find it more useful to discuss it.

1.4 DECOMPOSING GREAT AUNTS AND OTHER RELATIONS

The alert reader will realise that we are not really being ghoulish in giving this section its title – we are using the word 'decomposition' in the sense of the last paragraph.

We may see that there are *two* aspects to her moods that one might call the *physiological* and the *psychological*. She has three physiological states – asleep, drunk and sober – and two psychological states – in a pleasant or a foul temper. Each of her moods is a combination of two of these, as illustrated in Table 1.3.

Table 1.3.

Mood	*Physio-*	*Psycho-*
asleep	asleep	–
genial	sober	pleasant
irritable	sober	foul
maudlin	drunk	pleasant
unspeakable	drunk	foul

We may 'construct' Great Aunt Eugenia from two smaller and simpler 'Aunts', one of whom is concerned only with effects on her physiological state and the other of whom has only psychological states.

The two are connected 'in series'. Stimuli are presented to the first Aunt, the physiological Aunt, who changes state accordingly and sends an