NOR unit the extra time taken was small. To deal with functions of many more than 8 variables would probably have necessitated also writing array (i) on disc.

After compilation the amount of time taken for execution of the program was quite short, not being more than ten minutes for a net with 8 external inputs, and very much shorter for nets with a lesser number of external inputs.

Acknowledgement

The author would like to thank Mr. Brian Foster of IBM for help in writing the program.

Reference

QUINE, W. V. (1955). 'A way to simplify truth functions', *American Math. Monthly*, Vol. 62, No. 9, pp. 627–31.

Book Review

Sequential Machines and Automata Theory, by TAYLOR L. BOOTH, 1967; 592 pages. (New York, London, Sydney: *John Wiley and Sons Inc.*, 162s.)

According to the preface of this book it was written 'to provide a unified treatment of sequential machines and automata theory and their interrelationships' and uses 'an engineering rather than a formal mathematical style of presentation'.

What is 'engineering style'? Presumably it is a style of writing in which one is not presented with highly formal definitions and detailed rigorous proofs but a style which concentrates on basic ideas and motivations, which gives ideas of proofs which will stand up if one considers them more deeply and which does not formalise for the sake of formalisation. A superb book written in such a style and covering much (but not all) of the material in Booth's book is 'Computation, Finite and Infinite Machines', by M. Minsky (see review in this *Journal*, Vol. 10, p. 391). Unfortunately many of the informal statements of Booth, whilst it is easy for someone knowledgeable in the subject to see what he intends to say, are rather vague and all too easily collapse if examined closely. There should be more explanation of the concepts involved and less of a rush into formalisations in the 'engineering approach' of this book.

This is unfortunate as the book could have been very good indeed. It has wide coverage (mathematical preliminaries; finite state machines, their decomposition, minimalisation and identification; regular expressions; Turing machines; recursive functions and computability problems; phrase structure grammars; Markov processes and probabalistic machines). It is well planned with informal introductions, more formal definitions, relations to other systems and key properties, summaries, good bibliographies and examples. It has illustrative diagrams, charts and tables. It is a well produced book. Printing errors exist but are rare.

Some specific criticisms follow. The definitions of, and relations between, computable, partially computable, algorithm and program (p. 360) are vague. No mention is made of why such an apparently restrictive definition of 'computable' is used, i.e. of Church's Thesis. At the end of p. 361 the following sections are motivated as 'defining the relationship between Turing machines and computable functions'. As a computable function has been defined as one which a Turing machine can compute this is a strange motivation. On p. 407 it is stated that there is a strong connection between finite state machines and finite state languages because sentences of such languages are generated from left to right; this is vague, which would not matter so much except that attempts to make it precise could lead to incorrect conclusions. Generally the 'proofs' that a certain machine cannot perform a given task only consider one possible way the machine might work; a remark should be added that it can be proved that any way one thinks of will not work. Regular expressions are motivated as 'describing the behaviour of' sequential machines. How do they describe behaviour? Moreover sequential machines are defined as possibly having an infinite set of states whereas regular expressions are related to finite state automata. In the definition of semigroup on p. 32 the fact $c \epsilon S$ should not be a hypothesis and the equations $a\circ(b\circ c) = (a\circ b)\circ c = a\circ b\circ c$ are confusing without a comment that the last expression is merely an abbreviation for either of the first two. An example on p. 33 is described as 'illustrating the usefulness' of a certain procedure, but does no such thing. It is untrue that A. M. Turing was 'not interested in the design of information-processing devices' (p. 353).

These are all quibbles, but many such remarks could be made about statements in the book. The book is a useful one for a lecturer to possess, and to guide students in their reading of it, but not a book to be read without such guidance or without also referring to other books on the subject which give a better feel for the subject.

D. C. COOPER (Swansea)