Reviews of Papers in the Computer Field

D. L. Epley, Paper Reviews Editor
D. W. Fife, A. I. Rubin, R. A. Short, H. S. Stone
Assistant Paper Reviews Editors

Please address your comments and suggestions to the Paper Reviews Editor: Donald L. Epley, Department of Electrical Engineering, University of Iowa, Iowa City, Iowa 52240.

A. SWITCHING AND AUTOMATA THEORY


In the design of synchronous sequential machines, various canonical realizations which make use of feedback shift registers have been developed [1], [2]. This paper attempts to extend some of these results to realize asynchronous machines in a similar manner.

In an earlier paper by Brzozowski and Singh [3], in which canonical feedback-free realizations of definite machines1 were considered, the asynchronous unit delay (AUD) was introduced. The AUD is an n-input n-output asynchronous sequential circuit in which the present value of the output n-tuple is equal to the value of the input n-tuple before the last input change.

Extending the results of the earlier paper [3] it is shown that any fundamental mode asynchronous sequential machine M can be realized by a circuit of feedback index m with one \((n+m) \times (n+m)\) AUD element and m inertial delays2 where n is the number of binary inputs and m is the smallest integer not less than \(\log_2 s_k\) where \(s_k\) is the maximum number of stable states in any column of M. Realizations with the same feedback index have been obtained elsewhere [4], [5] and the relative virtues of these different realizations are debatable.

It is also shown that any fundamental mode asynchronous sequential machine M can be realized by an asynchronous circuit of feedback index 1 with 1 inertial delay and a chain of \(k(n+1) \times (n+1)\) AUD elements. Effect the chain of AUD elements operates as a synchronous shift register. The inertial delay elements are needed in order to prevent the circuit from malfunctioning due to races and hazards.

The paper is generally well written and has several examples which make it easy to read. As with the related material on synchronous realizations, the results seem to be primarily of theoretical rather than practical significance at this time.

It should be noted that the realization of the AUD from basic gate elements itself requires the presence of feedback. Hence the realizations considered in this paper could really best be compared with minimum feedback realizations of synchronous machines using memory elements more complex than delay elements, such as SR flip-flops, a problem which, incidentally, has not been solved.

A. D. Friedman
Bell Telephone Labs.
Murray Hill, N. J.

REFERENCES


The paper begins with a good formal introduction to iterative arrays, discussing briefly their relation to other automata, particularly the "tessellation structures" of Moore [1] and von Neumann [2]. Attention is then restricted to iterative arrays viewed as real-time tape acceptors. The author proves a speedup theorem, which shows how to speed up an array by a constant factor \(k\). The speedup is done by using a length \(k\) encoding of the input tapes, and realizing blocks of the array as finite-state machines in a new array which operates \(k\) times as fast. The complexity classes of arrays defined by the author ignore the complexity of the modules of which the array is composed. Thus, the speeded-up array is a member of the same class of arrays as the array whose behavior it imitates. The author then proves that the pattern of interconnection ("stencil") of any array may be reduced to allow direct communication only between nearest neighbors without reducing the real-time computing power of the array. This again involves increasing the complexity of the finite-state machines in the array.

Arrays to accept two languages (YT and palindromes) are presented, and it is shown that the context-free languages neither contain nor are contained in the set of languages accepted by real-time iterative arrays. The languages accepted are shown to be closed under union, intersection, and complementation, but not under reflection or concatenation. The author constructs a language \(X_k\) which he shows is not accepted by any \(n\)-dimensional (real-time) iterative array, and which he elsewhere [3] shows is accepted by an \((n+1)\) dimensional iterative array. Thus, a hierarchy of languages is established, based on the dimension of the iterative arrays which accept them.

While the speedup theorems the author obtains is of some interest, this result shares with many others of this type a certain lack of practical applicability. It can only be obtained by lumping together all finite-state automata without regard to their relative complexities. The stencil reduction result depends on this same lumping of automata. In this respect, the author's approach can be seen to be radically different from those of

1 A definite machine is a sequential machine in which the output is dependent only on the last \(k\) values of the input for some finite value of \(k\).
2 An inertial delay of magnitude \(D\) acts like a pure delay to pulses of width \(\geq D\) but does not respond to pulses of width \(< D\).
von Neumann [2], Codd [4], and others who attempt to achieve various computational goals using minimal (for example, fewest states) primitive modules. Input and output are all done from a single machine, which, while no theoretical limitation because of the lumping of finite-state machines, is a further dissimilarity between the author’s arrays and practical parallel-processing computing schemes.

Together, the results on the hierarchy (based on dimension) and the power of arrays with a simple stencil (only nearest neighbors) constitute a principle that having an extra dimension available is more powerful than merely having a broader reach in fewer dimensions. It might be interesting to see how this principle would be affected by placing various restrictions on the classes of finite-state machines in the array.

ERIK D. GOODMAN
Logic of Computers Group
University of Michigan
Ann Arbor, Mich.

REFERENCES

In summary, the paper is a rather elegant mathematical exercise and the results add greater significance to the work of Hartmanis and Stearns, Zeiger, and Krohn and Rhodes.

MONROE NEWBORN
Dept. of Elec. Engng.
Columbia University
New York, N. Y. 10027


H. Wang (“Circuit synthesis by solving sequential Boolean equations,” Zeit. für Math. Logik und Grundlagen der Math., vol. 5 pp. 291–322, 1959) pointed out that sequential Boolean equations, obtained by adding a sequential operator to take care of the time element to the ordinary Boolean operations, provided a convenient language for expressing conditions to be satisfied by circuits or automata. Among other things he described algorithms for deciding whether an equation has a solution, and for deciding whether it has a solution with finite delay. The present paper presents algorithms which are thought to be more efficient.

The basic equation considered is

\[ E : F(X, Y, dY) = 0. \]

Here \( X = (x_1(t), \ldots, x_n(t)) \), \( Y = (y_1(t), \ldots, y_m(t)) \) where the \( x_i(t), y_j(t) \) are Boolean-valued variables, \( dY(t) = Y(t+1) \), and \( F \) is a Boolean expression. It is said to have a solution if for every finite sequence \( X(1), \ldots, X(L) \) of values of \( X \) there exists a sequence of values for \( Y \), \( Y(1), \ldots, Y(L+1) \) such that \( E \) holds for all \( t = 1, 2, \ldots, L \). Similarly it is said to have a solution with delay \( d \) if the knowledge of \( X(1), \ldots, X(d) \) is sufficient to determine a value for \( Y(1) \). Thereafter, for all \( r \), knowledge of \( Y(r), X(r), X(r+1), \ldots, X(r+d) \) is sufficient to determine a value for \( Y(r+1) \) so that the determined part of the sequence for \( Y \) is a part of a solution for the given sequence of \( X \). The authors observe that the form \( E \) includes the apparently more general form \( G(X, Y, dX, dY) = 0 \) considered by Wang, and of course finite sets of such equations; also, as Wang observed, equations involving \( d^2 Y, d^3 Y, \) etc.

The first result is that the problem of deciding whether \( E \) has a solution is equivalent to the problem of deciding whether a finite labeled graph \( G \) is solvable, i.e., whether for every tape on the alphabet \( \Sigma \) of labels there exists a path on \( G \) whose sequence of label is the given tape. This is clearly equivalent to the problem of deciding whether a nondeterministic automaton accepts all tapes, which is easily solvable. For solvability with delay \( d \) the treatment is similar but more complicated. Bounds are obtained for the minimal possible delay which throw doubt on some of Wang’s results for this case. The paper contains a useful Appendix on the solution of ordinary Boolean equations.

J. C. SHEPHERDSOHN
University of Bristol
Bristol, England


This paper is concerned with the recognition of words which are contained in languages defined by pushdown store machines. Such languages exhibit many of the syntactic properties of algorithmic programming languages. Thus the recognition problem for languages generated by pushdown store machines is related to compilation of programming languages.

A pushdown automaton (PDA) consists of a finite state control which can scan an input tape and can read from and write onto a pushdown store. If the finite state control can scan the input tape in only one direction, it