A multitape one-way nonwriting automaton (MONA) is a finite state machine with a finite number of one-way input tapes which are advanced independently. This paper summarizes closure properties and decision problems of both deterministic and nondeterministic varieties. The family of \( n \)-ary word relations defined by deterministic (nondeterministic) \( n \)-tape MONA's is called \( D_n(N_n) \). Clearly \( D_n = N_1 \) is regular sets and requires no further comment.

As is often the case, the results for deterministic machines are more difficult and less general than those for nondeterministic machines. Indeed, if one attaches multiple input tapes to any AFA [1] or closed class of one-way nondeterministic balloon automata [3], one obtains at once a family \( L_w \) of \( n \)-ary relations with most of the properties of \( N_w \). Defining operations componentwise in the obvious manner, it turns out that any such \( L_n \) is a full AFL [1] (closed under union, concatenation, star, homomorphism, inverse homomorphism, and intersection with "regular" relations of the form \( R_1X \cdots XR_m \), each \( R_i \in N_1 \)). Any projection of a member of \( L_n \) on \( k \) coordinates clearly belongs to \( L_k \). Since each \( L_n \) must contain \( N_n \), the Rabin and Scott proof of the undecidability of the disjointness problem for two-tape deterministic MONA yields the undecidability of the universe problem for \( N_n \) and so \( L_n \), and hence all the usual undecidability results follow by appropriate generalization of methods of [2] including the fact that if the deterministic subfamily is properly contained in \( L_n \), it is undecidable whether a member of \( L_n \) can be defined by a deterministic machine.

In view of the connection between \( N_n \) and linear context-free languages established by Rosenberg [5], it is not surprising to find that the results on closure properties and decision problems for \( N_2 \) (and \( N_n \)) echo those for the context-free languages. Most of the results for \( P_2 \) are parallel to those for deterministic context-free languages (the connection is [5] does not extend in a natural way to the deterministic case); the equivalence problem for \( D_2 \) remains open. An interesting example of the pathology present even in as simple a family as \( N_2 \) is the table demonstrating the independence of the three properties: 1) \( L \) in \( D_2 \), 2) \( L^* \) in \( D_2 \) for all \( k \geq 1 \), and 3) \( L^* \) in \( D_2 \).

S. A. Greibach

Harvard University

Cambridge, Mass.

References


This paper is written by automata theorists for automata theorists. In numerous definitions of automata, there is an "input head" which is expected to be positioned at one square of an "input tape" at all times. There is often the possibility that the automaton might cause the input head to move right, say, from the rightmost input square, before the automaton has had a chance to do all the computation it wanted to do. Thus, "endmarkers"—special symbols that appear on the leftmost and rightmost input squares but are not otherwise considered part of the input—are often part of the automaton definition. This paper demonstrates that in several cases the endmarkers are unnecessary in that a class of automata with end markers recognizes only languages that are recognizable by the analogous classes of automata without one or both of the endmarkers.

The paper is divided into two sections. The first section concerns stack automata [2]. A stack automaton is a device with a two-way read-only input tape with endmarkers, a finite control, and a push-down stack which can be entered in a read-only mode by a "stack head." The authors show that the same languages are recognized if the stack automaton is redefined to have a right endmarker but no left one. The construction involved is of medium difficulty, and should be understood by a person who is familiar with stack automata and reads the short intuitive explanation given in the paper.

The second section concerns the elimination of endmarkers from the deterministic linear bonded automaton [1]. As the authors state, this result is somewhat in the public domain, as it has been taken for granted by those interested in the matter.

Unfortunately, the proof of this theorem consists mainly of a formal construction of an endmarker stack automaton simulating a two endmarker automaton. It consumes 63 pages, fully half the paper. Since those interested in the paper will have a good deal of "physical" intuition about stack automata, a formal construction that neglects this intuition cannot be the most efficient way to present a proof. Moreover, while a theorem may be of interest, it is often the ideas and algorithms involved in its proof that lead to other results. It is thus important that the thoughts which lead to the theorem be emphasized as well as the theorem itself. Lengthy formal constructions obscure these ideas.

There was, incidentally, a very good alternative to the construction given. In [3], a "high level programming language" for stack automata was designed, and the construction could be couched much more succinctly in this language with the same result. One would have to verify that the program was correct, but the correctness of the construction in question was not given either. Comments about [3] by Prof. Ginsburg [4] presenting the other side of the coin might be of interest to the reader.

J. D. Ullman

Bell Telephone Laboratories, Inc.

Murray Hill, N. J.

References


B. ANALOG AND HYBRID TECHNIQUES


This paper describes a special analog correlator which is used for optical character recognition (OCR). The correlator is built around a special "card capacity ready only storage (CCROS)" memory. Each bit in this CCROS memory is represented by a 1-pF capacitor, which may be removed from the circuit by appropriately punching a special Hollerith card on which is printed one plate of the capacitor. The other plate of the capacitor is printed on a printed circuit board pressed against the reverse side of the card, and the portion of the card between the two plates forms the capacitor's dielectric.

In the memory which was constructed, a single card contains a total of 720 capacitance plates arranged in 12 horizontal rows with 60 plates per row. The 60 plates on each row occupy column positions 11–70 inclusive of the Hollerith matrix. All plates in a row are connected together by a "word" line, also imprinted on the card. The 720 mating capacitance plates on the printed circuit board against which the card is held are connected together in vertical columns. Each line which connects the 12 plates in a column together is called a "bit" line. A single card with its mating array of capacitance plates on the printed circuit board then has 12 word lines and 60 bit lines with a 1-pF capacitor connected from each word line to each bit line. Punching the card at any specific location severs the connection between