Listed are abstracts from recent papers by IBM authors. Inquiries should be directed to the publications cited.

Abstracts

On Arithmetic Expressions and Trees, R. R. Redziejowski, *Communications of the ACM* 12, No. 2, 81–84 (February 1969). A description is given of how a tree representing the evaluation of an arithmetic expression can be drawn in such a way that the number of accumulators needed for the computation can be represented in a straightforward manner. This representation reduces the choice of the best order of computation to a specific problem under the theory of graphs. An algorithm to solve this problem is presented.

Bounded Action Machines: Toward an Abstract Theory of Computer Structure, E. G. Wagner, *Journal of Computer and System Sciences* 2, No. 1, 13–75 (June 1968). The axiomatically defined Bounded Action Machine is presented as an approach to an abstract theory of computer structure–organization–architecture. The basic property of a Bounded Action Machine is that there is a finite upper bound on the number of storage locations which can be active (contents "accessed or changed") at any step in the machine's operation. The paper starts with a precise formulation of this property within a very general framework. The study proceeds by investigating the consequences of imposing additional axioms on this basic structure, the new axioms being chosen so as to promote simpler and more uniform structure (i.e., so as to rule out "pathological structures"). This leads first to a characterization of "addressable memory." It is shown to be finite under very general conditions. The latter part of the paper is addressed to the structure of the "remainder" of the memory. This culminates in a general characterization of the concept of "tapes." Together these results constitute a general characterization of the existing computer specie (including Turing machines). This paper includes comparison of this model with those of other authors and a discussion of possible directions of future research.

Classes of Matrices with Distinct, Real Characteristic Values, C. E. Radke, *SIAM Journal of Applied Mathematics* 16, No. 6, 1192–1207 (November 1968). In the solution to sets of linear differential or difference equations it is desired to know whether or not the roots of the characteristic equation are distinct. Matrices often are used to describe such sets of linear equations and the characteristic equation of the describing matrix is also that for the set of equations. One class of matrices called the oscillatory matrices is known to have distinct, positive, and real characteristic values or roots of the characteristic equation. A class of matrices called the alternating matrices is introduced and is shown to have distinct and real characteristic values. An interesting subclass of the alternating matrices consists of (0, 1) matrices in which the positive elements form a triangular block array along the secondary diagonal. The characteristic values of this subclass, in addition to being distinct and real, are shown to possess a property of alternating sign as one progresses down the sequence of characteristic values as ordered in terms of magnitude. One such (0, 1) matrix which belongs to this subclass is shown to be the matrix which describes the set of difference equations obtained from the cascade merge-sort in data processing.

Counter Machines and Counter Languages, P. C. Fischer,* A. R. Meyer, and A. L. Rosenberg, *Mathematical Systems Theory* **2**, No. 3, 265–283 (September 1968). The languages recognizable by time- and space-restricted multiple-counter machines are compared to the languages recognizable by similarly restricted multiple-tape Turing machines. Special emphasis is placed on languages definable by machines which operate in "real time." Time and space requirements for counter machines and Turing machines are analyzed. A number of questions which remain open for time-restricted Turing machines are settled for their counter machine counterparts.

* Cornell University, Ithaca, New York.

Description of FORMAT, a Text-Processing Program, G. M. Berns, *Communications of the ACM* **12**, No. 3, 141–146 (March 1969). FORMAT is a production program which facilitates the editing and printing of "finished" documents directly on the printer of a relatively small (64k) computer system. It features good performance, totally free-form input, very flexible formatting capabilities including up to eight columns per page, automatic capitalization, aids for index construction, and a minimum of nontext items. It is written entirely in FORTRAN IV.

Designers Get New Way to Talk with Computers, M. A. Howard, *Product Engineering* **40**, No. 2, 76–78 (January 27, 1969). A Computer Application Service (CAPS), developed to help design and development engineers use a computer to solve mechanical design and analysis problems, is described. CAPS provides the engineer with comprehensive design and analysis assistance early in the development process to avoid functional problems and to provide more reliable operation when the design is put into hardware. CAPS also helps the engineer determine the effects of design changes on existing mechanisms. A paper feed design problem is used to illustrate how CAPS works.

A Note on Time-Sharing, D. Chazan, A. G. Konheim, and B. Weiss, *Journal of Combinatorial Theory* **5**, No. 4, 344–369 (December 1968). The setting for the problem discussed here is a service facility which is to be "time-shared" by two customers. A precise notion of a processing schedule, which prescribes the times at which the facility is available to each customer, is introduced. Associated with each schedule is the expected total waiting time of the two customers. The schedules which minimize this time are called optimum schedules and are determined here. A number of examples and extensions are given which indicate the scope of the methods used.