Abstracts of Current Computer Literature

Notice: These abstracts were prepared on a commercial basis by Information Associates under the direction of Dr. Geoffrey Knight, Jr. The complete material is unavailable from either the IEEE or Information Associates. If you require copies of any of the papers referred to, we suggest you contact the authors directly. All government research reports (those identified by CFSTI) can be obtained from Clearinghouse, Springfield, Va. 22151.

CONTENTS

ABSTRACTS ................................................................. Pages 115-132

Abstract Numbers

0) GENERAL; STANDARDS; EDUCATION .................. —
1) LOGIC AND SWITCHING THEORY; SEQUENTIAL MACHINES ................................................. 8210-8224
2) DIGITAL COMPUTERS AND SYSTEMS ................ 8225
3) LOGIC DEVICES AND CIRCUITS (HARDWARE) .... 8226
4) DIGITAL STORAGE AND INPUT-OUTPUT EQUIPMENT ....................................................... 8227-8230
5) PROGRAMMING OF DIGITAL MACHINES ............ 8231-8247
6) LINGUISTICS, DOCUMENTATION, AND HUMANITIES ...................................................... 8248
7) BEHAVIORAL SCIENCE, PATTERN RECOGNITION, AND ARTIFICIAL INTELLIGENCE ............ 8249-8258
8) MATHEMATICS .......................................................... 8259-8262
9) PROBABILITY, MATHEMATICAL PROGRAMMING, DIGITAL SIMULATION, INFORMATION THEORY, AND COMMUNICATION SYSTEMS ................................................. 8263-8279
10) SCIENCE, ENGINEERING, AND MEDICINE ......... 8280-8282
11) ANALOG AND HYBRID COMPUTERS .................. —
12) REAL-TIME SYSTEMS AND AUTOMATIC CONTROL: INDUSTRIAL APPLICATIONS ................. 8283-8284
13) GOVERNMENT, MILITARY, AND TRANSPORTATION APPLICATIONS .............................. —
14) BUSINESS APPLICATIONS OF INFORMATION PROCESSING ........................................... —

_DESCRIPTOR-IN-CONTEXT INDEX ........................................ Pages 124-129
_IDENTIFIER INDEX .......................................................... Pages 130-131
_AUTHOR INDEX .............................................................. Page 132
1) LOGIC AND SWITCHING THEORY; SEQUENTIAL MACHINES


Directed graphs having logical control associated with each vertex have been introduced as models of computational tasks for automatic assignment and sequencing on parallel processors. A brief review of their properties is given. A procedure to test the “legality” of graphs in this class is described, and leads to algorithms for counting the number of all possible executions (and-type subgraphs), and for evaluating the probability of ever reaching a given vertex in the graph. Numerical results are given for some example graphs.


This paper shows how a “unity-ratio” totally symmetric function can be identified without any further decomposition. The identification is carried out by writing the given function in its ordered partitioned tabular form. The method is general, straightforward, and programmable on a digital computer.


The original multivalued algebra defined by Post in 1921 included the cyclization operation in its generating set. Up to now the algebra has not been used for the synthesis of ternary digital systems because of two shortcomings: no simple canonical forms are possible, and the cyclization is difficult to implement. This correspondence presents an algebra that includes the cyclization operation (defined as a generalization of the cyclization) in its generating set together with three truncating operations. The set not only leads to simple, minimizable normal forms, but allows a special type of function decomposition. The minimization method and an algorithm for the composite functions are given and electronic circuits implementing the combinational and sequential logic are illustrated. Finally, as an example of the synthesis procedure, a fast carry-propagation adder of the Kilburn type is synthesized.


This paper describes a new approach to the design of combinational logic using large-scale integrated (LSI) circuit technology. A simple “prototype” logic function of n binary variables is imbedded within an array defined as a (n + 1)-dimensional affine group such that input variable encodings are not affected by feedback from the function’s output. This approach to logic design complements rather than replaces conventional multilevel logic design. The relative complexity is strongly dependent on the specific function (or set of functions) to be realized. In some cases, complexity is reduced; in others it is increased. Basically, exclusive-or gates have been introduced into the logic design in “bag of tricks” in an explicit and effective way: as an array rather than as separate components. This provides practical advantages, such as economical LSI array structures and effective and new computational tools for the logic designer. The number of prototypes required to generate all functions of n variables is equal to the number of “prototype” equivalence classes (or P-classes) into which the group of feedback-free affine transformations partitions the space of all n-input, single-output switching functions. For n = 3, 4, or 5, the required number of prototypes is 3, 8, or 48, respectively. As n becomes larger, the required number of prototypes increases to 2^2n - n, and the number of different functions that can be generated from a single randomly selected prototype approaches 22n - 1. All transformation groups which have previously appeared in the literature on combinational switching theory are subgroups of the group considered herein, and further subdivide the prototype classes. Computer programs have been written which identify the prototype associated with any given four-input function and almost all five-input functions. These programs have been used to find explicit prototype functions or canonical representatives for 46 out of the 48 equivalence classes for n = 5. The relative size of each prototype class has also been estimated.


This paper presents a canonical form for context-sensitive derivations and a parsing algorithm which finds each context-sensitive analysis once and only once. The amount of memory required by the algorithm is essentially no more than that required to store a single complete derivation. In addition, a modified version of the basic algorithm is presented which blocks infinite analyses for grammars which contain loops. The algorithm is also compared with several previous parsers for context-sensitive grammars and general rewriting systems, and the difference between the two types of analyses is discussed. The algorithm appears to be complementary to an algorithm by Kuno in several respects, including the space-time tradeoff and the degree of context dependence involved.


This paper is concerned with the class of “web grammars,” introduced by Pfaltz and Rosenberg, whose languages are sets of labelled graphs. A slightly modified definition of web grammar is given, in which the rewriting rules can have an applicability condition, and it is proved that, in general, this extension does not increase the generative power of the grammar. This extension is useful, however, for otherwise it is not possible to incorporate negative contextual conditions into the rules, since the context of a given vertex can be unbounded. A number of web grammars are presented which define interesting classes of graphs, including unseparable graphs, unseparable planar graphs, and planar graphs. All the grammars in this paper use “normal embeddings” in which the conditions between the web and the host web are conserved, so that any rewriting rule affects the web only locally.


A slip language is a language whose Parikh mapping is a semilinear set. A slip family is a family containing only slip languages. The purpose of the paper is to study slip AFL. A sufficiency condition is given on a slip family which ensures that the family generates a slip AFL. Using this condition, it is proved that 1) there exists the largest slip AFL that is a slip family, and 2) if L is a slip family, then the smallest AFL containing the commutative closure of L is a slip AFL. A new operation called “homomorphic replication” is then introduced. It is shown that the smallest AFL containing a homomorphic replication of a slip AFL is also a slip AFL. Furthermore, the resulting AFL is principal if the original AFL is principal. It is then proved that the smallest AFL containing all homomorphic replications of the regular sets is not principal. Finally, a restricted families of acceptors are presented which, respectively, define the smallest AFL containing a particular homomorphic replication of the regular sets and all homomorphic replications of the regular sets.


In this paper the necessary and sufficient conditions of representability of nonregular languages in finite probabilistic automata are formulated.


In this paper the algebraic treatment of probabilistic automata with two states is discussed. The authors derive the result that the matrix product corresponding to a given input tape can be decomposed into the sum of a finite number of fundamental matrices which are determined by the matrices corresponding to the input symbols. The new concept of probabilistic automata completely isolated by the L-th approximation is introduced. With respect to these automata, all tapes of length greater than or equal to L+1 can be classified into 2^L sets by means of their (L+1) suffixes. By using this concept, it is demonstrated that a two-input symbol actual automaton with two states can realize any definite event.

8219 Deciphering Automata in the Absence of an Upper Bound of the State Number, Y. M.
Stochastic Automaton Mode of Feature Selection in Pattern Recognition—see 8252.

8222


A sequential network is said to be controllable if there exists at least one integer $k$ such that it is possible to transition between any pair of arbitrary states $(S_i, S_j)$ in exactly $k$ steps. In this paper, necessary and sufficient conditions are given for a nonlinear sequential network to be controllable. Strong connectedness is a necessary condition for controllability. It is shown that the existence of two cycles $C_1$ and $C_2$ on a strongly connected sequential network, whose cycle lengths $L_2$ and $L_2$ are relatively prime, is both necessary and sufficient for controllability. Simple test procedures are also developed which determine if a sequential network is controllable and which determine the transition sequences.

8223


The synthesis of sequential discrete systems involves two major steps: 1) the abstract synthesis that yields a state-transition diagram, and 2) the structure synthesis that yields a realization. The structure synthesis, where the minimum state-transition diagram is assumed to be given, is discussed. Classically, two different models are in use to perform the structure synthesis: the finite-state model with memory span $k=1$, and the finite-memory model. These classical models (the finite-state model extended for $k>1$), are considered and other models are introduced that may be used for the structure synthesis. The latter, combined models with memory span $k (k>1)$, sometimes yield a better realization (according to specific engineering requirements) than the classical models. Structure synthesis based on the labeling models and a rough evaluation of all the considered models from the standpoint of their realization is included in this paper together with an outline of a FORTRAN IV program being developed to automate the complete structure synthesis.

Approximation of Information Channels by Generalized Deterministic Sequential Machines—see 8279.

8224


Methods already exist for the construction of critical race-free assignments for asynchronous sequential circuits. Some of these methods permit the construction of many assignments for the same flow table. The algorithm presented here consists of two easy to apply tests which select that critical race-free assignment most likely to produce a set of simple next-state equations. The algorithm has been programed.

2) DIGITAL COMPUTERS AND SYSTEMS

8225

Virtual Fourth Generation Computer, E. D. Callender (Aerospace Corp., San Bernardino); Rept. TR-0066(S999-1) SAMSO-TR-70-145, 17 pp., April 1970; CFSTI, AD 705 580, $3.00.

The document is a survey paper dealing with the logical form of the fourth generation computer. The notions of virtual machine, parallel structure, and hierarchical construction are blended together to create such a computer. This blend results in a very powerful and flexible approach to general purpose computers. The paper briefly considers both the hardware and software implications of such a design.

Computer Architecture for Process Control—see 8283.

Synthesis of a Fast Carry Propagation Adder Using Ternary Logic—see 8212.

Realization of Digital Filters Using Block Floating-Point Arithmetic—see 8268.

Legality and Other Properties of Graph Models of Computations in Parallel Processors—see 8210.

Model for Predicting the Performance of Computer Systems—see 8210.

Cyclic Algebra for Synthesizing Ternary Digital Systems—see 8212.

3) LOGIC DEVICES AND CIRCUITS (HARDWARE)

8226


Digital logic circuits are now available and are being used with delay times that are comparable to the delays of interconnections used in packaging these circuits. At high speeds, however, such interconnections no longer behave as simple short circuits, but take on the appearance of transmission lines. Unless transmission lines are terminated properly, "reflections" can develop that might be of sufficient magnitude to produce false logic levels or exceed maximum circuit voltage specifications. One may choose to solve the problem by increasing the density of the system. This, however, introduces the problem of "crosstalk." This article describes several analytical techniques for predicting the kinds of reflections and crosstalk that are typically seen in digital systems, thus enabling the engineer to determine in advance whether or not such "interconnection noise" will result, how bad it will be, and what the typical interconnection limitations are for circuits of various speeds.

Circuits Implementing Combinational and Sequential Ternary Logic—see 8212.

Fourier Transform Approach to the Design of Combinational Logic Using Large-Scale Integrated Circuit Technology—see 8213.

4) DIGITAL STORAGE AND INPUT–OUTPUT EQUIPMENT

8227

Application of Partial-Response Channel Coding to Magnetic Recording Systems, H. Kobayashi

A magnetic recording channel can be regarded as a “partial-response” channel because of its inherent differentiation in the readback process. The conventional NRZI method of recording as described above is equivalent to the “encoding” of this particular partial-response channel, the purpose of which is to limit the propagation of error in the channel output. Using this new viewpoint, one can readily adopt an error-detection scheme (developed for general partial-response channels) that takes full advantage of the inherent redundancy in the three-level channel output. The detection scheme is optimum in the sense that it detects all detectable errors with minimum delay. The paper also describes a new high-density recording method, named the “interleaved NRZI,” which is obtained by molding an ordinary recording channel into a different type of partial-response channel, resulting in a potential increase in channel capacity. Implications of the corresponding optimum error detection scheme is also presented. Finally, performance of these error detection schemes is evaluated in terms of probabilities of detecting single and double errors within a certain finite delay.

**8228**


The man–machine interface at any terminal in a computer system is a likely source of error and can be regarded as a noisy channel. Certain data, such as ID numbers, can be protected to protect against most likely errors, including transportation of adjacent symbols and substitutions as well as deletions and insertions. This paper first considers certain basic requirements for error detection with minimum redundancy. An efficient special coding scheme designed for decimal terminals is described next. Finally, certain cyclic codes are shown to be adaptable to transposition error control when appropriate decoding schemes are implemented.

Software and Hardware for Conversational On-Line Design of Control Systems—see 8284.

**8229**


The effects of several parameters of symbol formation on the legibility of CRT displays have been studied. These parameters were letter orientation, slanted versus upright; letter generation method, dot matrix versus stroke; dot matrix size, 5 x 7 and 7 x 9; and dot geometry, elongated versus circular dots. Twenty-six letters and ten numbers comprised the symbol set that was tested. The 5 x 7 and 7 x 9 symbols were designed for legibility while the stroke symbols were patterned after the LED symbol. Symbols were tested by presenting them briefly on the screen of the display with subjects attempting to identify the symbol shown. Speed and accuracy of identification were used as the measures of legibility. Results indicate that 7 x 9 dot matrix symbols drawn with circular dots were superior to all other symbols in both reaction time and error measures. Slanting had a detrimental effect on dot and stroke symbols and circular dot symbols were superior to elongated dot symbols.

**8230**  
Computer Processing for Display of Three-Dimensional Structures, R. B. Desens (Naval Postgraduate School, Monterey); 228 pp., October 1969; CFSTI, AD 706 018, $7.00.

The field of computer graphics applied to three-dimensional space is introduced through a discussion of perspective transformations, data structure, contour lines, and the problem of hidden-line removal. The transformation of three-dimensional coordinates into two-dimensional picture-plane coordinates is developed for twelve degrees of freedom, allowing the simultaneous movement and rotation of both the object under view and the observer. Basic concepts and requirements for the structure of data and ideas for the use of contour lines are discussed as a relative part of the field of three-dimensional computer graphics. An algorithm for the removal of hidden lines is explained for the case where the objects under view can be assumed to be constructed of bounded plane surfaces.

5) PROGRAMMING OF DIGITAL MACHINES

**8231**


The problems of convergence, correctness, and equivalence of computer programs can be formulated by means of the satisfiability or validity of certain first-order formulas. An algorithm is presented for constructing such formulas for functional programs, i.e., programs defined by LISP-like conditional recursive expressions.

Software Implications of a Virtual Fourth Generation Computer With Parallel Structure and Hierarchical Organization—see 8225.

Software for Conversational On-Line Design of Control Systems—see 8284.

**8232**

JOSTRAN: An Interactive JOSS Dialect for Writing and Debugging FORTRAN Programs, W. R. Graham and D. C. MacNeillage (Rand Corp., Santa Monica); Rept. RM-6248-PR, 14 pp., March 1970; CFSTI, AD 704 568, $3.00.

The document gives a description of JOSTRAN, a JOSS dialect that expedites the construction of FORTRAN programs. JOSS is an interactive, on-line computer system. JOSS-language programs are list-processed; i.e., each statement is interpreted at execution time. FORTRAN is the principal language for programming digital computers to perform numerical calculations. The JOSS language permits greater flexibility and subtlety, but FORTRAN can handle larger calculations. JOSTRAN, a specified FORTRAN-compatible dialect of JOSS, combines the advantages of both languages. It allows the user to exploit JOSS's interactive, list-processing facilities while writing and debugging a program, and facilitates the translation of the JOSTRAN program into FORTRAN.

**8233**


In this paper trade-offs among certain computational factors in hash coding are analyzed. The paradigm problem considered is that of testing a series of messages one-by-one for membership in a given set of messages. Two new hash-coding methods are examined and compared with a particular conventional hash-coding method. The computational factors considered are the size of the hash area (space), the time required to identify a message as a nonmember of the given set (reject time), and an allowable error frequency. The new methods are intended to reduce the amount of space required to contain the hash-coded information from that associated with conventional methods. The reduction in space is accomplished by exploiting the possibility that a small fraction of errors of commission may be tolerable in some applications, in particular, applications in which a large amount of data is involved and a core resident hash area is consequently not feasible. In such applications, it is envisaged that overall performance could be improved by using a smaller core resident hash area in conjunction with the new methods and, when necessary, by using some secondary, and perhaps time-consuming test to “catch” the small fraction of errors associated with the new methods. An example is discussed which illustrates possible areas of application for the new methods. Analysis of the paradigm problem demonstrates that allowing a small number of test messages to be falsely identified as members of the given set will permit a much smaller hash area to be used without increasing reject time.

**8234**


A general method of file structuring is proposed which uses a hashing function to define file structure. Two types of such sets are examined, and their relation to trees studied in the past is explained. Results for the probability distributions of path lengths are derived and illustrated.

**8235**

An Implementation of LISP 1.5 for the IBM 360/67 Computer, D. G. Gentry (Naval Postgraduate School, Monterey); 99 pp., December 1969; CFSTI, AD 706 031, $3.00.

The design and implementation of the NPS LISP programming system is described. NPS LISP is an interactive version of LISP 1.5, a sophisticated list processing and symbol manipulation computer language. NPS LISP was implemented in PL/I for operation under the CP/MICOS time-sharing system on the IBM 360/67 computer. It is an interpretive system patterned after 7090 LISP. Most of the features of 7090 LISP are included in NPS LISP.

**8236**


STAGE 2 is the second level of a bootstrap sequence which is easily implemented on any computer. It is a flexible, powerful macroprocessor designed specifically as a tool for constructing machine-independent software. In this paper the features provided by STAGE 2 are sum-
marized, and the implementation techniques which have made it possible to have STAGE2 running on a new machine with less than one man-week of effort are discussed. The approach has been successful on over 15 machines of widely varying characteristics.


A programming system using a hypothetical computer is proposed for use in teaching machine and assembly language programming courses. Major components such as monitor, assembler, interpreter, grader, and diagnostics are described. The interpreter is programmed and documented for use on an IBM 360/67. The interpreter can be used for teaching machine language programming and can be incorporated into the proposed programming system.


The optimization of memory hierarchy involves the selection of types and sizes of memory devices such that the average access time to an information block is a minimum for a particular cost constraint. It is assumed that the frequency of usage of the information is known a priori. In this paper the optimization theory for a single task or program is reviewed and it is extended to a general case in multiprogramming when a number of tasks are executed concurrently. Another important extension treats the case when memories are available only in indivisible modules. Comparisons with conventional methods of solution as well as computational experience on the multiprogrammed and modular cases are given.

8239 An Operating System for the PDP-8/I, J. S. Heidt, C. L. Fricks, W. D. Smith, and T.-Y. Wang (Georgia Inst. of Tech., Atlanta); GITTIS-70-03, 73 pp., 1970; CSFTI, PB 190 816, $3.00.

An operating system for the PDP-8/I, specifically designed around hardware operating characteristics and providing interrupt handling facilities and task supervision capabilities, is discussed. Several alternative methods of implementation and possible future expansions are analyzed. The operating system's flowcharts and coding in SL/8, an intermediate language designed for the project, are included.

8240 SL/8: A Synthesis Language for the PDP-8/I, J. S. Heidt and C. L. Fricks (Georgia Inst. of Tech., Atlanta); GITTIS-70-02, 29 pp., 1970; CSFTI, PB 190 818, $3.00.

SL/8 is an intermediate language designed for the purpose of writing an operating system for the PDP-8/I. It provides the convenience of a higher-level language and the capabilities of machine language. A modified version, users SL/8, is suitable as a user language for programs running under the operating system.

8241 Simulation of Interference of Packets in the ALOHA Time-Sharing System, W. H. Bortels (Hawaii U., Honolulu); Rept. TH MIS-

The information-gathering aspect of sorting is considered from a theoretical viewpoint. A large class of algorithms for sorting is defined, based on the idea of information use. Properties of this algorithm class are developed, and it is noted that several well-known sorting algorithms are closely related to algorithms in R. The binary tree sort is shown to be in R and to have unique properties in this class. A vector is defined which characterizes the information-gathering efficiency of the algorithms in R. Finally, a more general class of algorithms is defined, and some of the definitions extended to this class. Two intrinsically nonrandom algorithms are given which appear to require graph theory or combinatorial topology for their solution.


The methods currently in use and previously proposed for the choice of a root in minimal storage tree sorting are in reality methods for making inefficient statistical estimates of the median of the sequence to be sorted. By making efficient use of the information in a random sample chosen during input of the sequence to be sorted, significant improvements over ordinary minimal storage tree sorting can be made. A procedure is proposed which is a generalization of minimal storage tree sorting and which has the following three properties: 1) There is a significant improvement (over ordinary minimal storage tree sorting) in the expected number of comparisons required to sort the input sequence. 2) The procedure is statistically insensitive to bias in the input sequence. 3) The expected number of comparisons required by the procedure approaches (slowly) the information-theoretic lower bound on the number of comparisons required. The procedure is, therefore, "asymptotically optimal."


It is shown that, owing to certain restrictions placed upon the set of admissible structures, some previous solutions have not characterized trees in which expected search time is minimized. The more general problem is shown to be a special case of a coding problem, which was previously formulated and solved as a linear integer programming problem, and in the special case of equally probable key requests is found to be solvable almost by inspection. Some remarks are given regarding the possibility of realizing a shorter computational procedure than would be expected from an integer programming algorithm, along with a comparison of results from the present method with those of the previous.


A syntax-directed picture analysis system based on a formal picture description scheme is described. The system accepts a description of a set of pictures in terms of a grammar generating strings in a picture description language; the grammar is explicitly used to direct the analysis or parse, and to control the calls on pattern classification routines for primitive picture-com-
ponents. Pictures are represented by directed graphs with labeled edges, where the edges denote elementary picture components and the graph connectivity mirrors the picture component connectivity; blank and don’t care “patterns” allow the description of simple relations between visible patterns. The bulk of the paper is concerned with the picture parsing algorithm which is an n-dimensional analog of a classical top-down string parser, and an application of an implemented system to the analysis of spark chamber film. The potential benefits of this approach, as demonstrated by the application, include ease of implementation and modification of picture processing systems, and simplification of the pattern recognition problem by automatically taking advantage of contextual information.

Web Grammars for Picture Interpretation—see 8215.

Data Structures for Three-Dimensional Graphical Processing—see 8220.

6) LINGUISTICS, DOCUMENTATION, AND HUMANITIES

8248
Real English—A Description of its Operation, D. Klappholz (Penn U., Philadelphia); Rept. 70-26 AROD-5208-6-RT, 23 pp., March 1970; CFSTI, AD 705 651, $3.00.

Real English is a translator from English to a symbolic, computer-oriented language which allows its user to query a data bank in the language most natural to him. Real English was designed to operate in a time-sharing environment, in conjunction with a generalized information retrieval system. It consists of a syntactic component, which provides a “string analysis” of an input request, a semantic component which determines what information the user is requesting, and a command formatting component, which composes symbolic information retrieval requests of the sort accepted by the information retrieval system.

Storage and Retrieval Efficiency Using New Methods of Hash Addressing—see 8233.

More Efficient Searching Than by Using Integer Programming—see 8246.

On-Line Display Editing of Text Strings Using Preamarked Randomly Addressable Magnetic Tapes—see 8243.

7) BEHAVIORAL SCIENCE, PATTERN RECOGNITION, AND ARTIFICIAL INTELLIGENCE

8249

An algorithm is presented which partitions a given sample from a multimodal fuzzy set into unimodal fuzzy sets. It is proven that if certain assumptions are satisfied, then the algorithm will derive the optimal partition in the sense of maximum separation. The algorithm is applied to the problem of clustering data, defined in multidimensional space, into homogeneous groups; it is applied to the classification of experimental data and results and errors are discussed in detail. Methods for extending the algorithm to the clustering of very large sets of points are also described. The advantages of the method (as a clustering technique) are that it does not require large blocks of high speed memory, the amount of computing time is relatively small, and the shape of the distribution of points in a group can be quite general.

8250

This paper describes an on-line interactive graphics system which has been designed to solve the problems of pattern analysis and pattern classification. A wide variety of both classical and unique mathematical algorithms, along with their graphic system implementation, are discussed. A discussion of an application of the system to the handprinted character recognition problem is included.

8251

The selection of a “best” subset of features from a given set is considered for the case where a multiplicity of constraints are present in the two-class pattern recognition problem. The problem can be formulated with a Lagrangian multiplier and an efficient iterative technique used in conjunction with dynamic programming procedures to obtain a solution when two constraints are present. The problem can be formulated in integer programming terms and use made of available integer programming computer algorithms for solution when multiple constraints are present. The Lemke-Spiegelman zero-one integer algorithm is particularly useful, but other integer programming algorithms can be used also.

8252

The problem of feature selection in pattern recognition is briefly reviewed. Feature selection techniques discussed include 1) information theoretic approach, 2) direct estimation of error probability, 3) feature-space transformation, and 4) approach to using stochastic automata model. Computer simulation results are presented and compared.

8253

A linear classifier based on linear programming which is adaptive to a change in the set of input vectors is discussed. Different from other linear classifiers, this one maintains the maximum reliability of its operation, provided that the set of pattern vectors is linearly separable. A procedure of deriving an optimum structure of the linear classifier for a change in the set of input vectors is presented. An application of the ordinary simplex method and yields an optimum structure in much fewer iterations than the straightforward application of the ordinary simplex method does. The adaptive procedure is then extended to the case in which a linear classifier maintains the minimum number of erroneously classified input vectors even if the set of input pattern vectors is not linearly separable. This is based on Gomory’s algorithm for integer linear programming. The feasibility and efficiency of these linear classifiers are computationally proved by some examples.

8254

Pattern classification can be considered as consisting of two parts: 1) pattern detection—the process of learning from a set of sample patterns of known classifications and discriminating characteristics of each category; and 2) actual classification—the process of recognizing patterns of unknown classifications as members of particular categories. The paper is a study in the first part of the process since it is most often the more important of the two parts of the pattern classification scheme. An algorithm for establishing decision criteria of classification is described. Evaluation is made on its performance, computation time, and data storage requirement.

8255

A nonparametric training procedure capable of processing an arbitrary sequence of patterns based on stochastic approximation techniques is considered. An acceleration scheme based on the adaptive Robbins–Monro procedure to increase the rate of convergence and to make the estimation asymptotically efficient is proposed. This scheme has been applied to the problem of recognition of handwritten characters.

8256

The Cyclops research project is aimed at developing computer-based systems that perform various processes on pictorial information, including enhancement and recognition. The work described in the report uses this pre-processing system as the basis for further research in the domain of recognition. Through this research, a process for matching fingerprints was developed. With this process, fingerprints are preprocessed to extract certain minutiae, and these minutiae are then matched to determine if the fingerprints match. The present report describes the matching programs in detail. A sample run of the process for three fingerprints is shown.

8257

A new technique is given for establishing the completeness of resolution-based deductive systems for first-order logic (with or without equality) and several new completeness results.
are proved using this technique. The technique leads to very simple and clear completeness proofs and can be used to establish the completeness of most resolution-based deductive systems reported in the literature. The main new result obtained by means of this technique is that a linear format for resolution with merging and set of support and with several further restrictions is a complete deductive system for the first-order predicate calculus.


The resolution principle is an inference rule for quantifier-free first-order predicate calculus. In the past, the completeness theorems for resolution and its refinements have been stated and proved for finite sets of clauses. It is easy (by Godel's compactness theorem) and of practical interest to extend them to countable sets, thus allowing schemata representing denumerably many axioms. In addition, some theorems similar to Craig's interpolation theorem are proved for deduction by resolution. In propositional calculus, the theorem proved is stronger, whereas in predicate calculus the theorems proved are in some ways stronger and in some ways weaker than Craig's theorem. These interpolation theorems suggest procedures which could be embodied in computer programs for automatic proof finding and consequence finding.

8) MATHEMATICS


A new version of the Euclidean algorithm for finding the greatest common divisor of n integers a_i and multipliers x_i such that gcd a_1 x_1 + ... + a_n x_n is presented. The number of arithmetic operations and the number of storage locations are linear in n. A theorem of Lamé that gives a bound for the number of iterations of the Euclidean algorithm for two integers is extended to the case of n integers. An algorithm to construct a minimal set of multipliers is presented. A FORTRAN program for the algorithm appears as Comm. ACM Algorithm 386.

Cyclic Algebra for Synthesizing Ternary Digital Systems—see 8212.


The nonlinear interpolation of functions of very many variables is discussed. Deterministic termwise assessment of a prohibitively large number of terms naturally leads to a choice of random sampling of these numerous terms. After introduction of an appropriate higher order interpolation formula, a working algorithm is established by the Monte Carlo method. Numerical examples are also given.


With the recent appearance of efficient quadratic programming algorithms, the well-known "magnified diagonal" method of nonlinear regression is now easily extended to include linear inequality constraints. One such extension is proposed and some of its properties are discussed.

Nonlinear Regression Analysis by Time-Sharing Computers—see 8282.


Differential equations are developed for the smoothing density function and for the smoothed expectation of an arbitrary function of the state. The exact equations developed are difficult to solve except in trivially simple cases. Approximations to these equations are developed for the smoothed expectation of the state and the smoothing covariance matrix. For linear systems these equations reduce to previously derived results. An iterative technique is suggested for even greater accuracy in approximations for severely nonlinear systems.

Nonlinear Interpolation of Multivariable Functions by the Monte Carlo Method—see 8260.


A direct design procedure for nonrecursive digital filters, based primarily on the frequency-response characteristic of the desired filters, is presented. An optimization technique is used to minimize the maximum deviation of the synthesized filter from the ideal filter over some frequency range. Using this frequency-sampling technique, a wide variety of low-pass and band-pass filters have been designed, as well as several wide-band differentiators. Some experimental results on truncation of the filter coefficients are also presented. A brief discussion of the technique of nonuniform sampling is also included.


A practical method is described for designing recursive digital filters with arbitrary, prescribed magnitude characteristics. The method uses the Fletcher-Powell optimization algorithm to minimize a square-error criterion in the frequency domain. A strategy is described whereby stability and minimum-phase constraints are observed, while still using the unconstrained optimization algorithm. The cascade canonical form is used, so that the resultant filters can be realized accurately and simply. Design examples are given of low-pass, wide-band differentiator, linear discriminator, and vowel formant filters.

8268 Realization of Digital Filters Using Block-Floating-Point Arithmetic, A. V. Oppenheim (MIT,
More Efficient Searching Than by Using Integer Programming—see 8246.

Simulation of Interference of Packets in a Time-Sharing System Using Random Access Communication—see 8241.

Legality and Other Properties of Graph Models of Computations in Parallel Processors—see 8210.

Stochastic Automaton Model of Feature Selection in Pattern Recognition—see 8252.

Methods are presented for the encoding of information into binary sequences in which the number of zeros occurring between each pair of successive ones has both an upper and a lower bound. The techniques, based on the state structure of the constraints, permit the construction of short, efficient codes with favorable error-propagation-limiting properties.


A new class of multiple-error correcting codes has been developed. Since it belongs to the class of one-step-decodable majority codes, it can be decoded at an exceptionally high speed. This class of codes is derived from a set of mutually orthogonal Latin squares. This mutually orthogonal property provides a class of codes having a unique feature of “modularity.” The parity check matrix possesses a uniform pattern and results in a small number of inputs to modulo-2 adders. This class of codes has \( m^2 \) data bits, where \( m \) is an integer, and 2\( m \) check bits for \( t \)-error correcting.


The class of codes described in this paper is used for single-error correction and double-error detection (SEC-DED). It is equivalent to the Hamming SEC-DED code in the sense that for a specified number \( k \) of data bits, the same number of check bits \( r \) is used. The minimum odd-weight-column code is suitable for applications to computer memories or parallel systems. A computation indicates that this code is better in performance, cost, and simplicity than are conventional Hamming SEC-DED codes.


A high-speed method is derived for single-symbol error correcting Reed-Solomon and Hamming type codes. A matrix description is used for implementation of the codes, in which single-error correction in the Galois field \( 2^b \) corresponds to correcting a block of \( b \) bits in a binary field. The resulting codes correct not only single-bit errors but also single clusters of \( b \)-adjacent-bit errors.


Three measures of the complexity of error correcting decoders are considered, namely, logic complexity, computation time, and computational work (the number of logical operations). Bounds on the complexity required with each measure to decode with probability of error \( P_e \) at code rate \( R \) are given and the complexity of a number of ad hoc decoding procedures is examined.

8277 Maximal Group Codes with Specified Minimum Distance, A. M. Patel (IBM, Poughkeepsie);
ABSTRACTS OF CURRENT COMPUTER LITERATURE


All $n$-digit maximal block codes with a specified minimum distance $d$ such that $2d > n$ can be constructed from the Hadamard matrices. These codes meet the Plotkin bound. In this paper all maximal group codes in the region $2d > n$ are shown. The modular terms $N_2(n,d)$ are defined by $N_2(n,d) = \sum_{i=0}^{d/2} N_i(n,d)$. It is shown that the value of $N_2(n,d)$ largely depends on the binary structure of the number $d$. An algorithm is developed that determines $N_2(n,d)$, the maximum number of code words for given $d$ and $n \leq 2d$. The maximal code is then, given by its modular representation, explicitly in terms of certain binary coefficients and constants related to $n$ and $d$. As a side result, a new upper bound on the number of code words in the region $2d \leq n$ is obtained which is, in general, stronger than Plotkin's extended bound.

Application of Partial-Response Channel Coding to Magnetic Recording Systems—see 8227.

Coding for Minimum Redundancy Error Detection at the Man-Machine Interface of a Computer System—see 8228.

8278

A comparison is made of the performance of pure retransmission, forward error correction, and hybrid (error detecting/correcting) schemes for data transmission in a noisy (probability of error, $P = 10^{-4}$) binary symmetric channel. The performance calculations are based on the use of BCH codes for error detection and correction. In this case, the full correction capability of the code is used. A probability of undetected error of less than $10^{-9}$ error/bit, can be achieved by correcting only a few errors while retaining a reasonable throughput. A hybrid scheme of low retransmission rate. The best codes in the class considered are specified and the complexity of instrumentation is estimated. Finally, various combinations of possible systems employing half duplex and reverse channel operation are used in a comparison of the transmission schemes. For line error rate worse than $10^{-4}$ error/bit, a hybrid system operating with a reverse channel is superior to the other possibilities.

8279

In the article, the following approximation problem is discussed: Let $N_i(X, Y)$ be the set of all information channels with the output alphabet $X$ and the input $Y$. Let $\rho$ be a distance function defined for every $v, v \in N_i(X, Y)$ as $\rho(v, w) = \sum |x \cdot v - y \cdot w|$, where the sum is taken over all $v$ and $w$, $x_0 \cdot x_0$, ..., $y_0 \cdot y_0$, ..., $x_0 \cdot x_0$. Let $N = N_i(X, Y)$ be the set of all weakly stationary regular information channels with finite past history, and let $N_2 = N_i(X, Y)$ be the set of all channels realized by random (=probabilistic) automata. Then $N_2$ is dense in $(N, \rho)$. This theorem has an important consequence that every channel from $N$ can be approximated by a channel realized by a given binary stationary independent source and suitable generalized sequential deterministic machines.

10) SCIENCE, ENGINEERING, AND MEDICINE

8280

The history of the long lines computer project is mentioned briefly and is followed by more detail of the feasibility and implementation stages. The system aims for the project are outlined and then introduced as a carefully staged implementation program. The descriptions of Stage I, due to be introduced at once in 1970, gives the facilities to be offered, discusses the record updating problems during the changeover to computer working, and outlines the arrangements made to cover the possible failure of the computer. The article concludes that the project should be implemented as soon as possible but stresses the need to proceed carefully to ensure success.

8281

Scrpps Institution of Oceanography is currently using three IBM 1800 digital computers for shipboard oceanographic research. These digital computers are medium-sized computers capable of time-sharing several levels of programs on a priority basis. In this sense the computers can be considered as small computer centers; scientific programs which are batch-process oriented can be operated at the same time as continuously running data logging programs, control programs, and other dynamic type programs. Two distinct types of program thus evolve: process and nonprocess types. Process programs are those programs operating and/or interacting with external elements in near real-time, plus the bookkeeping programs supporting this level of activity. Nonprocess programs are those more normally seen in any scientific computer center. The data they use are acquired from previously stored sources (punched cards, magnetic tape), and they generally analyze and change data into some form more amenable to scientific study. There are several unique aspects to the Scrpps seagoing computer centers; each computer is both a process control type computer and a batch process type computer.

Statistical Testing of Random Number Genera
tion for Nuclear Monte Carlo Programs—see 8263.

Pattern Recognition Applied to Fingerprint Matching—see 8256.

8282

Analysis of biomedical data frequently involves obtaining a mathematical equation to describe numerical data. Because of the nature of biological systems, the desired equations are often not linear. In the past, such nonlinear systems were often avoided because mathematical methods for analyzing them were tedious and, occasionally, quite imprecise. Modern computers, have made such systems far easier to work with. The publication describes a program for obtaining these nonlinear "regression" equations. The program described has a number of advantageous properties: 1) it can be used for almost any type of equation, including exponential, logarithmic, and power functions; 2) it is written in the basic program language, which can be learned by a person without computer knowledge in only a few hours; 3) it is designed for time-sharing computers, so that the user can readily interact with the computer and modify the equations being used at will; and 4) the output of the program includes sufficient statistical parameters to enable further evaluation of the results. In addition to giving the program, the publication gives full instructions on its use, suggestions for modifying the program for additional applications, and several illustrative examples.

12) REAL-TIME SYSTEMS AND AUTOMATIC CONTROL: INDUSTRIAL APPLICATIONS

8283

The process control requires that a digital computer's resources be shared among several functional tasks. A method is outlined which accomplishes this sharing through the use of a hardware executive controller and hardware memory boundaries. This approach guarantees that all scheduled tasks will be completed and that all programs and data will be protected.

8284

The digital computer with its enormous capacity for numerical problems and great speed of operation, is ideally suited to the handling of the many routine calculations which arise in the design of multivariable systems. In this role, it can relieve the designer of this necessary but tedious part of the design process. However, as it is not possible to program the "insight" and "intuition" of the designer, the digital computer can best be considered as a valuable design aid. The basic philosophy for a conversational on-line design facility using video-graphic aids is presented. The software structure is outlined and the hardware requirement is discussed.
**Expanded Index**

### A

**Access** Simulation of Interference of Packets in a Time-Sharing System Using Random Access Communication 8241

Operating System for Conversational Access to a 2048-Word Computer 8243

**Accuracy**

- see Error

**Adaptive**

Adaptive Linear Classifier for Pattern Recognition Based on Linear Programming 8253

- see also Control, Learning, Patterns, Training

**Addition**

Synthesis of a Fast Carry Propagation Adder Using Ternary Logic 8212

**Adders, Addressing**

Storage and Retrieval Efficiency Using New Methods of Hash Addressing 8233

On-Line Display Editing of Text Strings Using Premarked Randomly Addressable Magnetic Tapes 8243

- see also Storage

**Algebra; Algebraic**

Cyclic Algebra for Synthesizing Ternary Digital Systems 8212

Algebraic Treatment of Two-State Symbol Probabilistic Automata 8218

- see also Automata, Boolean, Matrices, Sequential Machines

**Algorithms; Algorithmic**

Information Gathering Efficiency of Sorting Algorithms 8244

Error Propagation Properties Associated with High-Radix FFT Algorithms for Implementation of Nonrecursive Filters 8269

**Algorithms For:**

Counting the Number of All Possible Executions in Parallel Processors 8210

Minimization of Normal Forms and Determinism of Composite Functions in Ternary Algebra 8212

Selection of Encoding Transformations to Realize Switching Functions from Their Prototypes 8213

Context-Sensitive Parsing 8214

State Assignment Selection in Asynchronous Sequential Circuit Realization 8224

Solution of the Hidden-Line Problem 8230

Construction of Formulas for Formulating the Problems of Convergence, Correctness, and Equivalence of Functional Programs 8231

Efficient Operation of On-Line Tape Systems 8242

Sampling Approach to Minimal Storage Tree Sorting 8245

More Efficient Searching Than by Using Integer Programming 8246

Parsing of Graph Representable Pictures 8247

Detection of Unimodal Fuzzy Sets 8249

Interactive Pattern Analysis and Classification 8250

Selection of Pattern Features by Mathematical Programming 8251

Pattern Classification with a Partitioned Training Set 8254

Finding the Greatest Common Divisor for n Integers 8259

**Approximate Solution of Free Boundary Problems for Elliptic Differential Equations Using Finite Differences** 8261

Magnetized Diagonal Method of Nonlinear Regression with Linear Constraints Using Quadratic Programming 8264

Optimum Design of Nonrecursive Digital Filters Using a Frequency-Sampling Method 8266

Computer-Aided Design of Recursive Digital Filters 8267

Maximal Group Codes with Specified Minimum Distance 8277

- see also Resource Management

**Analog Computers; Analog Components**

Initial-Value Theory for Fredholm Integral Equations with Semidegenerate Kernels for Digital and Analog Computer Solution 8262

**Analysis; Analytic**

Interactive Pattern Analysis and Classification 8250

Analysis of the Effectiveness of Error Detecting and Correcting Codes for Data Transmission in a Noisy Channel 8278

Nonlinear Regression Analysis by Time-Sharing Computers 8282

- see also Simulation, Syntax

**Approximation; Curve Fitting**

Pattern Classification Using Stochastic Approximation Techniques 8255

Approximate Solution of Free Boundary Problems for Elliptic Differential Equations Using Finite Differences 8261

Iterative Technique for Accurate Smoothing Approximations in Nonlinear Systems 8265

Approximation Problem for Nonrecursive Digital Filters 8266

Approximation of Information Channels by Generalized Deterministic Sequential Machines 8279

- see also Error, Interpolation

**Arithmetic**

Realization of Digital Filters Using Block Floating Point Arithmetic 8268

- see also Addition, Floating Point

**Arrays**

- see Cells, Storage

**Artificial Intelligence**

- see Learning, Patterns, Theorem Proving

**Assemblers; Assembly**

Monitor, Assembler, Interpreter, Grader, and Diagnostics for Teaching Machine Language Programming 8237

On-Line System Providing Text Editing, Automatic Filing and File Maintenance, and Program Preparation and Assembly 8243

**Automata; Turing Machines**

Algebraic Treatment of Two-State Symbol Probabilistic Automata 8218

Deciphering Automata in the Absence of an Upper Bound of the State Number 8219

Theory of Interacting Local Automata (Labeled Monadic Algebra) 8220

State Minimization of Nondeterministic Finite Automata 8221

Stochastic Automata Model of Feature Selection in Pattern Recognition 8252

- see also Context-Free, Languages, Machines, Sequential Machines

**Automated; Automatic; Automation**

Models for Automatic Assignment and Sequencing of Computational Tasks in Parallel Processors 8210

On-Line System Providing Text Editing, Automatic Filing and File Maintenance, and Program Preparation and Assembly 8243

Automatic Theorem Proving in First-Order Predicate Calculus 8257

Theorems to be Embodied in Computer Programs for Automatic Theorem Proving and Consequence Finding 8258

- see also Machines, Programming

**Binary; Nonbinary**

Properties of the Binary Tree Sort Algorithm 8244

- see also Addition, Arithmetic

**Boolean**

Identification of Totally Symmetric Boolean Functions 8211

- see also Sequential, Switching Functions

**Boundary; Boundaries**

Approximate Solution of Free Boundary Problems for Elliptic Differential Equations Using Finite Differences 8261

**C**

**Calculus**

- see Differentiation, Predicate Calculus

**Cells; Cellular**

Cellular Logic Arrays Using EXCLUSIVE-OR Gates 8213

**Characters**

- see Patterns, Recognition, Symbolic

**Circuits**

- see Combinational, Gates, Integrated Circuits, Logic, Modules, Sequential Machines

**Classification**

- see Patterns

**Codes; Coding; Decoding; Encoding**

Selection of Encoding Transformations to Realize Switching Functions from Their Prototypes 8213

Application of Partial-Response Channel Coding to Magnetic Recording Systems 8227

Coding for Minimum Redundancy Error Detection at the Man-Machine Interface of a Computer System 8228

Space/Time Tradeoffs in Hash Coding with Allowable Errors 8233

Coding Schemes for Reduction of Intersymbol Interference in Data Transmission Systems 8270

Pseudorandom Data Transmission Codes 8271

Sequence-State Methods for Run-Length-Limited Coding 8272

Orthogonal Latin-Square Codes for Correcting Multiple Errors 8273

Class of Optimal Minimum Odd-Weight-Column Single Error Correction and Double Error Detection Codes 8274

B-Adjacent Error Correcting Codes 8275

Measures of Error Correcting Decoder Complexity 8276

Maximal Group Codes with Specified Minimum Distance 8277
ANALYSIS OF THE EFFECTIVENESS OF ERROR DETECTING AND CORRECTING CODES FOR DATA TRANSMISSION IN A NOISY CHANNEL 8278

COMBINATIONAL; COMBINATORIAL
Circuit Implementing Combinational and Sequential Ternary Logic 8212
Fourier Transform Approach to the Design of Combinational Logic Using Large-Scale Integrated Circuit Technology 8213
Graph Theory and Combinatorial Analysis of Sorting 8244
—see also Graph Theory

COMMUNICATIONS; COMMUNICATIONS
Simulation of Interference of Packets in a Time-Sharing System Using Random Access Communication 8241
—see also Man–Machine Communications
Complexity
Measures of Error Correcting Decoder Complexity 8276

COMPUTATION; COMPUTING
Legality and Other Properties of Graph Models of Computations in Parallel Processes 8210
—see also Complexity

COMPUTATIONAL LINGUISTICS
—see Languages

COMPUTER APPLICATIONS
Computer-Aided Design of Recursive Digital Filters 8267
Implementation of a Telephone Long Lines Computer Project 8280
Programming of Time-Shared Computer Centers for Shipboard Oceanographic Research 8281
—see also Analog Computers, Automata, Computer Systems, Computers, Control, Editing, Patterns, Programs For, Retrieval, Simulation, Statistics

COMPUTER CENTER MANAGEMENT
Programming of Time-Shared Computer Centers for Shipboard Oceanographic Research 8281

COMPUTER SYSTEMS
Model for Predicting the Performance of Computer Systems 8210
Coding for Minimum Redundancy Error Detection at the Man–Machine Interface of a Computer System 8228
Hardware and Software Implications of a Virtual Fourth Generation Computer with Parallel Structure and Hierarchical Organization 8225
Proposed Programming System for the MIX Computer 8237
Operating System for the PDP 8/1 Computer Providing Interrupt Handling and Task Supervision Capabilities 8239
Operating System for Conversational Access to a 2048-Word Computer 8243
Computer Architecture for Process Control 8283
—see also Analog Computers, Automata, Computer Applications, Computer Center Management, Computer Systems, Digital Computers, Graphs, Processes, Sequential Machines, Time-Sharing

CONTEXT-FREE; CONTEXT-SENSITIVE
Context-Sensitive Parsing 8214

CONTROL
Directed Graphs Having Logical Control Associated with Each Vertex 8210
Controllability of Nonlinear Sequential Networks 8222
Error Control for Terminals with Human Operators 8228
Grammar for Controlling the Calls for Pattern Classification Routines for Picture Processing 8247
Computer Architecture for Process Control 8283
Software and Hardware for Conversational On-Line Design of Control Systems 8284

CONVERSATIONAL
Interactive Time-Sharing Command Generating Facility for Conversational Control of Tasks 8242
Operating System for Conversational Access to a 2048-Word Computer 8243
Software and Hardware for Conversational On-Line Design of Control Systems 8284
Correction
—see Error
Counters; Counting
Method of Counting the Number of All Possible Executions in Parallel Processes 8210

DATA
—see Data Bases, Data Structures, Files, Information, Lists, Sharing, Statistics, Trees

DATA Bases
Translator Routine for Querying Data Banks in a Time-Sharing Environment 8248

DATA PROCESSING
—see Computer Systems, Digital Computers, Processing

DATA STRUCTURES
Data Structures for Three-Dimensional Graphical Processing 8230
Graphical Data Structure Analysis 8250
—see also Lists

DATA TRANSMISSION
—see Codes, Communication, Interfaces
Debugging
Interactive joss Dialect for Writing and Debugging FORTRAN Programs 8232

DEDUCTION; DEDUCTIVE
Linear Format for Resolution with Merging in Deductive Systems for First-Order Logic 8257

DELAY
Digital Logic Circuits with Delay Times Comparable to the Delays of Interconnections Used in Packaging Them 8226

DESIGN
Optimum Design of Nonrecursive Digital Filters Using a Frequency-Sampling Method 8266
Computer-Aided Design of Recursive Digital Filters 8267
Software and Hardware for Conversational On-Line Design of Control Systems 8284
—see also Logic Design, Realization, Synthesis

DETECTION; DETECTORS
Optimum Error Detection in Magnetic Recording Channels 8227
Coding for Minimum Redundancy Error Detection at the Man–Machine Interface of a Computer System 8228
Detection of Unimodal Fuzzy Sets 8249
Class of Optimum Minimum Odd-Weight-Column Single Error Correction and Double Error Detection Codes 8274
Analysis of the Effectiveness of Error Detecting and Correcting Codes for Data Transmission in a Noisy Channel 8278

DIAGNOSIS
Monitor, Assembler, Interpreter, Grader, and Diagnostics for Teaching Machine

LANGUAGE PROGRAMMING 8237
—see also Error, Tests

DIFFERENTIAL EQUATIONS (PARTIAL)
—see Boundary

DIFFERENTIATION
Design of Wide-Band Differentiators 8266, 8267

DIGITAL; DECIMAL
Reflection and Crosstalk in Digital Logic Circuit Interconnections 8226
Simulation of the Performance of Decimal and Alphanumeric Codes 8270
—see also Communication, Control, Filtering, Integrated Circuits, Simulation

DIGITAL COMPUTERS; DIGITAL SYSTEMS
Cyclic Algebra for Synthesizing Ternary Digital Systems 8212
Initial-Value Theory for Fredholm Integral Equations with Semidegenerate Kernels for Digital and Analog Computer Solution 8262
Hardware Approach to Sharing a Digital Computer’s Resources for Process Control 8210
—see also Computer Systems, Computers, Processing

DISPLAYS
Effects of Parameters of Symbol Formation on the Legibility of CRT Displays 8229
Computer Processing for Display of Three-Dimensional Structures 8230
On-Line Display Editing of Text Strings Using Premarked Randomly Addressable Magnetic Tapes 8243
—see also Graphics, Terminals

DOCUMENTATION
—see Editing, Information, Retrieval

EDITING; EDITORS
On-Line System Providing Text Editing, Automatic Filing and File Maintenance, and Program Preparation and Assembly 8243

EFFICIENCY
Storage and Retrieval Efficiency Using New Methods of Hash Addressing 8233
Efficient Operation of On-Line Tape Systems 8243
Information Gathering Efficiency of Sorting Algorithms 8246
More Efficient Searching Than By Using Integer Programming 8246
Asymptotic Efficiency of Stochastic Approximation Techniques in Pattern Classification 8255
—see also Performance

ELECTRON BEAMS; CATHODE RAYS
Effects of Parameters of Symbol Formation on the Legibility of CRT Displays 8229

EQUATIONS
—see Boolean, Integral Equations

EQUIVALENCE
Equivalence of Automata and Reduced Automaton Matrices 8221
Construction of Formulas for Formulating the Problems of Convergence, Correctness, and Equivalence of Functional Programs 8231

ERROR
Optimum Error Detection in Magnetic Recording Channels 8227
Error Control for Terminals with Human Operators 8228
Space/Time Tradeoffs in Hash Coding with Allowable Errors 8233
Error Propagation Properties Associated
Evaluation—see Computation, Performance, Tests

Files; Filing
File Structures Using Hashing Functions to Define Tree Structures 8234
On-Line System Providing Text Editing, Automatic Filing and File Maintenance, and Program Preparation and Assembly 8243
Tree Structures for Optimal File Searching and Information Retrieval 8246
—see also Databases, Sorting

Filtering; Filters
Approximation Problem for Nonrecursive Digital Filters 8266
Computer-Aided Design of Recursive Digital Filters 8267
Realization of Digital Filters Using Block Floating-Point Arithmetic 8268
Fast Fourier Transform Organizations for High-Speed Digital Filtering 8269

Floating Point
Realization of Digital Filters Using Block Floating-Point Arithmetic 8268

Formal
Formalization of Properties of Functional Programs 8231
Syntax-Directed Picture Analysis System Based on a Formal Picture Description Scheme 8247
—see also Languages

Format; Formatting
Linear Format for Resolution with Merging in Deductive Systems for First-Order Logic 8257

Fourier
Fourier Transform Approach to the Design of Combinational Logic Using Large-Scale Integrated Circuit Technology 8213
Fast Fourier Transform Organizations for High-Speed Digital Filtering 8269

Functional; Functionals
Formalization of Properties of Functional Programs 8231

Functions
Minimization of Normal Forms and Determination of Composite Functions in Terminology Algebra 8212
File Structures Using Hashing Functions to Define Tree Structures 8234
Nonlinear Interpolation of Multivariable Functions by the Monte Carlo Method 8260
—see also Approximation, Boolean, Functional, Switching Functions

Gates
Cellular Logic Arrays Using EXCLUSIVE-OR Gates 8213

Generation; Generators
Interactive Time-Sharing Command Generating Facility for Conversationally Controlled Tasks 8242
Grammar Which Generates Strings in a Picture Description Language 8247
—see also Displays, Noise, Numbers

Graph Theory; Graphs
Legality and Other Properties of Graph Models of Computations in Parallel Processors 8210

Web Grammars Whose Languages are Sets of Labeled Graphs 8215
Graph Theory and Combinatorial Analysis of Sorting 8244
Parsing of Graph Representable Pictures 8247
—see also Trees

Graphics; Graphical
Web Grammars for Picture Interpretation 8215
Data Structures for Three-Dimensional Graphical Processing 8230
Parsing of Graph Representable Pictures 8247
On-Line Interactive Graphics System for Pattern Analysis and Classification 8250
—see also Displays, Patterns

Hardware
Hardware and Software Implications of a Virtual Fourth Generation Computer with Parallel Structure and Hierarchical Organization 8225
Hardware Approach to Sharing a Digital Computer's Resources for Process Control 8283
Software and Hardware for Conversational On-Line Design of Control Systems 8284
—see also Computers, Displays, Software, Storage

Hierarchical; Hierarchies
Hardware and Software Implications of a Virtual Fourth Generation Computer with Parallel Structure and Hierarchical Organization 8225
Optimization of Memory Hierarchies in Multiprogrammed Systems 8238

Human Factors
Error Control for Terminals with Human Operators 8228
—see also Man-Machine Communications

Information
Information Gathering Efficiency of Sorting Algorithms 8244
Information Theory Approach to Feature Selection in Pattern Recognition 8252
Approximation of Information Channels by Generalized Deterministic Sequential Machines 8279
—see also Codes, Retrieval

Integer Programming; Discrete Programming
Integer Programming Optimization of Modular Storage Hierarchies 8238
More Efficient Searching Than by Using Integer Programming 8246
Selection of Pattern Features by Integer Programming Algorithms 8251

Integral Equations
Initial-Value Theory for Fredholm Integral Equations With Semidegenerate Kernels for Digital and Analog Computer Solution 8262

Integrated Circuits; Microelectronics
Fourier Transform Approach to the Design of Combinational Logic Using Large-Scale Integrated Circuit Technology 8213
—see also Cells

Interaction
—see Graphics, Man-Machine Communications

Interconnections
Digital Logic Circuits with Delay Times Comparable to the Delays of Interconnections Used in Packaging Them 8226
—see also Integrated Circuits, Interfaces

Interpolations
Interpolation Theorems for Resolution in Lower Predicate Calculus 8258
Nonlinear Interpolation of Multivariable Functions by the Monte Carlo Method 8260

Interpreters; Interpretive
Interactive Interpretive List Processing and Symbol Manipulation Language 8235
Monitor, Assembler, Interpreter, Grader, and Diagnostics for Teaching Machine Language Programming 8237

Iterative; Iteration
Iterative Technique for Accurate Smoothing Approximations in Nonlinear Systems 8265

L
Languages; Linguistics; Grammars
Web Grammars Whose Languages are Sets of Labeled Graphs 8215
Abstract Family of Languages with the Semilinear Property 8216
Representability of Nongraphic Languages in Finite Probabilistic Automata 8217
Monitor, Assembler, Interpreter, Grader, and Diagnostics for Teaching Machine Language Programming 8237
Grammar Which Generates Strings in a Picture Description Language 8247
—see also Context-Free, Formal, Programming Languages, Sets, Syntax

Learning
Pattern Classification with a Partitioned Training Set 8254
—see also Adaptive

Linear Programming
Adaptive Linear Classifier for Pattern Recognition Based on Linear Programming 8253
—see also Integer Programming

Lists; List Processing
Use of List Processing Techniques in Writing and Debugging FORTRAN Programs 8232
Interactive Interpretive List Processing and Symbol Manipulation Language 8235

Logic; Logical
Directed Graphs Having Logical Control Associated with Each Vertex 8210
Circuits Implementing Combinational and Sequential Ternary Logic 8212
Reflection and Crosstalk in Digital Logic Circuit Interconnections 8226
Linear Format for Resolution with Merging in Deductive Systems for First-Order Logic 8257
—see also Cells, Combinational, Gates, Logic Design, Predicate Calculus, Sequential, Switching Functions

Logic Design
Fourier Transform Approach to the Design of Combinational Logic Using Large-Scale Integrated Circuit Technology 8213

M
Machines; Mechanization
Macroprocessor for Constructing Machine-Independent Software 8236
Monitor, Assembler, Interpreter, Grader, and Diagnostics for Teaching Machine
Language Programming 8237
—see also Automata, Computers, Man–Machine Communications, Sequential Machines

Macros
Implementation of a Mobile Macroprocessor Programming System 8236

Magnetic Recording
Application of Partial-Response Channel Coding to Magnetic Recording Systems 8227

Magnetic Tapes
On-Line Display Editing of Text Strings Using Premarked Randomly Addressable Magnetic Tapes 8243
—see also Magnetic Recording

Man–Machine Communications (Interaction, Systems)
Coding for Minimum Redundancy Error Detection at the Man–Machine Interface of a Computer System 8228
Interactive IDS Dialect for Writing and Debugging FORTRAN Programs 8232
Interactive Interpretive List Processing and Symbol Manipulation Language 8235
Interactive Pattern Analysis and Classification 8250
—see also Conversational, Displays, Graphics, Human Factors, Terminals

Management
—see Computer Center Management, Files, Resource Management

Mathematical Programming
Selection of Pattern Features by Mathematical Programming 8251
Magnified Diagonal Method of Nonlinear Regression with Linear Constraints Using Quadratic Programming 8264
—see also Integer Programming, Linear Programming, Optimal

Mathematics; Mathematical
—see Algebra, Arithmetic, Boolean, Differentiation, Fourier, Graph Theory, Logic, Models, Optimization, Probability, Statistics, Theorem Proving

Matrices
Equivalence of Automata and Reduced Automaton Matrices 8221

Measure; Measurement
Measures of Error Correcting Decoder Complexity 8276
—see also Analysis

Memory
—see Storage

Merging
Linear Format for Resolution with Merging in Deductive Systems for First-Order Logic 8257

Minimal; Minimization; Extrema
Minimization of Normal Forms and Determination of Composite Functions in Ternary Algebra 8212
State Minimization of Nondeterministic Finite Automata 8221
Coding for Minimum Redundancy Error Detection at the Man–Machine Interface of a Computer System 8228
Sampling Approach to Minimal Storage Tree Sorting 8245
Maximal Group Codes with Specified Minimum Distance 8277
—see also Optimal, Reduction, Switching Functions

Models; Modeling
Linguistics; Other Properties of Graph Models of Computations in Parallel Processors 8210
Multimodel and Computer Oriented Methodology for Synthesis of Sequential Discrete Systems 8223
Statistical Models of the Effects of Roundoff Noise in Digital Filters 8268
—see also Simulation

Modular; Modules; Building Blocks
Integer Programming Optimization of Modular Storage Hierarchies 8238
Modular Error-Correcting Codes 8273

Monitor Systems
Monitor, Assembler, Interpreter, Grader, and Diagnostics for Teaching Machine Language Programming 8237
Multiple-Access; Multi-User
—see Multiprogramming, Time-Sharing

Multiprogramming
Optimization of Memory Hierarchies in Multiprogrammed Systems 8238
—see also Time-Sharing

N

Networks; Nets
—see Modules, Sequential

Noise
Man–Machine Interface at a Terminal in a Computer System Considered as a Noisy Channel 8228
Statistical Models of the Effects of Roundoff Noise in Digital Filters 8268
Analysis of the Effectiveness of Error Detecting and Correcting Codes for Data Transmission in a Noisy Channel 8278
—see also Stochastic

Numbers; Numeric; Number Theory
Finding the Greatest Common Divisor for n Integers 8259
—see also Stochastic

Numerical Analysis
—see Algebra, Arithmetic, Differentiation, Functions, Integral Equations, Matrices

O

On-Line
On-Line System Providing Text Editing, Automatic Filing and File Maintenance, and Program Preparation and Assembly 8243
On-Line Interactive Graphics System for Pattern Analysis and Classification 8250
Software and Hardware for Conversational On-Line Control of Control Systems 8284

Operating Systems
Operating System for the PDP 8/1 Computer Providing Interrupt Handling and Task Supervision Capabilities 8239
Operating System for Conversational Access to a 2048-Word Computer 8243
—see also Monitor Systems

Operations Research
—see Mathematical Programming, Resource Management

Optimal; Optimization
Optimum Error Detection in Magnetic Recording Channels 8227
Optimization of Memory Hierarchies in Multiprogrammed Systems 8238
Tree Structures for Optimal File Searching and Information Retrieval 8246
Optimum Design of Nonrecursive Digital Filters Using a Frequency-Sampling Method 8266
Class of Optimal Minimum Odd-Weight-Column Linear Error Correction and Double Error Detection Codes 8274
—see also Linear Programming, Mathematical Programming, Minimal

Organization
—see Computer Systems, Computers

P

Parallel; Parallelism
Legality and Other Properties of Graph Models of Computations in Parallel Processors 8210
Hardware and Software Implications of a Virtual Fourth Generation Computer with Parallel Structure and Hierarchical Organization 8225

Parse
Context-Sensitive Parsing 8214
Parsing of Graph Representable Pictures 8247
—see also Languages

Patterns
Grammar for Controlling the Calls for Pattern Classification Routines for Picture Processing 8247
Clustering Technique for Pattern Recognition 8249
Interactive Pattern Analysis and Classification 8250
Selection of Pattern Features by Mathematical Programming 8251
Review of Methods of Feature Selection in Pattern Recognition 8252
Adaptive Linear Classifier for Pattern Recognition Based on Linear Programming 8253
Pattern Classification with a Partitioned Training Set 8254
Pattern Classification Using Stochastic Approximation Techniques 8255
Pattern Recognition Applied to Fingerprint Matching 8256
—see also Learning

Performance
Model for Predicting the Performance of Computer Systems 8210
Simulation of the Performance of Decimal and Alphanumeric Codes 8270
Analysis of the Effectiveness of Error Detecting and Correcting Codes for Data Transmission in a Noisy Channel 8278
—see also Computer Systems, Efficiency

Predicate Calculus
Automatic Theorem Proving in First-Order Predicate Calculus 8257
Interpolation Theorems for Resolution in Lower Predicate Calculus 8258

Probability; Probabilistic
Representability of Nonregular Languages in Finite Probabilistic Automata 8217
Algebraic Treatment of Two-State Two-Symbol Probabilistic Automata 8218
Sampling Approach to Minimal Storage Tree Sorting 8245
—see also Statistics, Stochastic

Processes; Processing; Processors
Computer Processing for Display of Three-Dimensional Structures 8230
Implementation of a Mobile Macroprocessor Programming System 8236
—see also Computers, Information, Lists, Macros, Parallel

Programming
Monitor, Assembler, Interpreter, Grader, and Diagnostics for Teaching Machine Language Programming 8237
Programming of Time-Shared Computer Centers for Shipboard Oceanographic Research 8281
—see also Mathematical Programming, Multiprogramming, Software
Programming Languages
Interactive 1088D Dialect for Writing and Debugging FORTRAN Programs 8232
Interactive Interpretive List Processing and Symbol Manipulation Language 8235
Language for Writing an Operating System for the PDP-11 Computer 8240
—see also Languages

Programming Systems
Time-Sharing Implementation of a LISP Programming System for the IBM 360/67 Computer 8235
Implementation of a Mobile Macroprocessor Programming System 8236
Proposed Programming System for the MIX Computer 8237

Programs; Routines
Formalization of Properties of Functional Programs 8231
On-Line System Providing Text Editing, Automatic Filing and File Maintenance, and Program Preparation and Assembly 8243
Grammar for Controlling the Calls for Pattern Classification Routines for Picture Processing 8247
Translator Routine for Querying Data Banks in a Time-Sharing Environment 8248
Theorems to be Embodied in Computer Programs for Automatic Theorem Proving and Consequence Finding 8258
Statistical Testing of Random Number Generation for Nuclear Monte Carlo Programs 8263
—see also Debugging, Software

Programs For:
Identification of Totally Symmetric Boolean Functions 8211
Selection of Encoding Transformations to Realize Switching Functions from Their Prototypes 8213
Automatic Structure Synthesis for Realizing Sequential Discrete Systems 8223
State Assignment Selection in Asynchronous Sequential Circuit Realization 8224
Finding the Greatest Common Divisor of n Integers 8259
Approximate Solution of Free Boundary Problems for Elliptic Differential Equations Using Finite Differences 8261
Random Number Generation 8263

Proof — see Theorem Proving

Random
On-Line Display Editing of Text Strings Using Premarked Randomly Addressable Magnetic Tapes 8243
Sampling Approach to Minimal Storage Tree Sorting 8243
Nonlinear Interpolation of Multivariable Functions by the Monte Carlo Method 8260
Random Number Generation 8263
—see also Access, Noise, Numbers, Stochastic

Realization; Implementation
Circuits Implementing Combinational and Sequential Ternary Logic 8212
State Assignment Selection in Asynchronous Sequential Circuit Realization 8224
Time-Sharing Implementation of a LISP Programming System for the IBM 360/67 Computer 8235
Implementation of a Mobile Macroprocessor Programming System 8236
Implementation of an Interactive Command Generating Facility Using a String Processing Language 8242
Realization of Digital Filters Using Block Floating-Point Arithmetic 8268
Error Propagation Properties Associated with High-Radix FFT Algorithms for Implementation of Nonrecursive Filters 8269
Implementation of a Telephone Long Lines Computer Project 8280
—see also Design, Synthesis

Recognition
Recognition Algorithm for Context-Sensitive Grammars 8214
—see also Patterns

Recursion; Recursive; Recurrence
Properties of Programs Defined by LISP-Like Conditional Recursive Expressions 8231
Approximation Problem for Nonrecursive Digital Filters 8266
Computer-Aided Design of Recursive Digital Filters 8267
Error Propagation Properties Associated with High-Radix FFT Algorithms for Implementation of Nonrecursive Filters 8269

Reduction
Coding Schemes for Reduction of Inter-Symbol Interference in Data Transmission Systems —see also Minimal

Redundancy
Coding for Minimum Redundancy Error Detection at the Man-Machine Interface of a Computer System 8228
Reliability — see Diagnosis, Errors, Redundancy

Resource Management
Hardware Approach to Sharing a Digital Computer's Resources for Process Control 8283

Retrieval
Storage and Retrieval Efficiency Using New Methods of Hash Addressing 8233
Tree Structures for Optimal File Searching and Information Retrieval 8246
Information Retrieval Using "Real English" 8248

Search —see Files, Retrieval

Sequences; Sequencing
Models for Automatic Assignment and Sequencing of Computational Tasks in Parallel Processors 8210
Nonparametric Training Procedure Capable of Producing an Arbitrary Sequence of Patterns 8255
Simulation of Uniform Random Sequences 8263
Sequence-State Methods for Run-Length-Limited Coding 8272
—see also Stochastic

Sequential
Circuits Implementing Combinational and Sequential Ternary Logic 8212
Multimodel and Computer Oriented Methodology for Synthesis of Sequential Discrete Systems 8223
—see also Sequential Machines

Sequential Machines (Circuits, Logic, Networks)
Controllability of Nonlinear Sequential Networks 8222
State Assignment Selection in Asynchronous Sequential Circuit Realization 8224
Approximation of Information Channels by Generalized Deterministic Sequential Machines 8279
—see also Automata, Machines

Sets
Web Grammars Whose Languages Are Sets of Labeled Graphs 8215
Detection of Unimodal Fuzzy Sets 8249

Sharing
Hardware Approach to Sharing a Digital Computer's Resources for Process Control 8283
—see also Time-Sharing

Simulation; Simulators
Simulation of Interference of Packets in a Time-Sharing System Using Random Access Communication 8241
Simulation of Uniform Random Sequences 8263
Simulation of the Performance of Decimal and Alphanumeric Codes 8270
—see also Models

Software
Hardware and Software Implications of a Virtual Fourth Generation Computer with Parallel Structure and Hierarchical Organization 8225
Macroprocessor for Constructing Machine-Independent Software 8236
Software and Hardware for Conversational On-Line Design of Control Systems 8284
—see also Monitor Systems, Operating Systems, Programming, Programming Systems, Programs

Sorting
Information Gathering Efficiency of Sorting Algorithms 8244
Sampling Approach to Minimal Storage Tree Sorting 8245

States
Sequence-State Methods for Run-Length-Limited Coding 8272
—see also Automata, Sequential Machines

Statistics; Statistical
Statistical Testing of Random Number Generation for Nuclear Monte Carlo Programs 8263
Magnified Diagonal Method of Nonlinear Regression with Linear Constraints Using Quadratic Programming 8264
Statistical Models of the Effects of Roundoff Noise in Digital Filters 8268
Nonlinear Regression Analysis by Time-Sharing Computers 8282
—see also Probability, Stochastic

Stochastic
Stochastic Automaton Model of Feature Selection in Pattern Recognition 8252
Pattern Classification Using Stochastic Approximation Techniques 8255
—see also Noise, Probability, Random

Storage; Memory
Storage and Retrieval Efficiency Using New Methods of Hash Addressing 8233
Optimization of Memory Hierarchies in Multiprogrammed Systems 8238
Sampling Approach to Minimal Storage Tree Sorting 8245
—see also Electron Beams, Files, Integrated Circuits, Magnetic Tapes, Retrieval

Strings
Implementation of an Interactive Command Generating Facility Using a String Processing Language 8242
On-Line Display Editing of Text Strings Using Premarked Randomly Addressable Magnetic Tapes 8243
Grammar Which Generates Strings in a Picture Description Language 8247
Structure
Automatic Structure Synthesis for Realizing Sequential Discrete Systems 8223
Hardware and Software Implications of a Virtual Fourth Generation Computer with Parallel Structure and Hierarchical Organization 8225
Computer Processing for Display of Three-Dimensional Structures 8230
—see also Data Structures, Files, Hierarchical, Lists

Switching Functions
Selection of Encoding Transformations to Realize Switching Functions from Their Prototypes 8213
—see also Automata, Boolean, Logic, Sequential Machines

Symbolic; Symbols
Effects of Parameters of Symbol Formation on the Legibility of CRT Displays 8229
Interactive Interpretive List Processing and Symbol Manipulation Language 8235

Syntax
Syntax-Directed Picture Analysis System Based on a Formal Picture Description Scheme 8247
—see also Languages

Synthesis
Cyclic Algebra for Synthesizing Ternary Digital Systems 8212
Multimodel and Computer Oriented Methodology for Synthesis of Sequential Discrete Systems 8223
—see also Design, Realization

Systems
Multimodel and Computer Oriented Methodology for Synthesis of Sequential Discrete Systems 8223
Iterative Technique for Accurate Smoothing Approximations in Nonlinear Systems 8265


Teaching
Monitor, Assembler, Interpreter, Grader, and Diagnostics for Teaching Machine Language Programming 8237
—see also Learning

Telephones
Implementation of a Telephone Long Lines Computer Project 8280

Terminals
Error Control for Terminals with Human Operators 8228
—see also Displays, Man-Machine Communications, Telephones

Ternary
Cyclic Algebra for Synthesizing Ternary Digital Systems 8212
Pseudoternary Data Transmission Codes 8271

Tests
Statistical Testing of Random Number Generation for Nuclear Monte Carlo Programs 8263
—see also Debugging, Diagnosis

Textual Data Processing
On-Line System Providing Text Editing, Automatic Filing and File Maintenance, and Program Preparation and Assembly 8243

Theorem Proving
Automatic Theorem Proving in First-Order Predicate Calculus 8257

Theorems to be Embodied in Computer Programs for Automatic Theorem Proving and Consequence Finding 8258

Time-Sharing
Time-Sharing Implementation of a LISP Programming System for the IBM 360/67 Computer 8235
Simulation of Interference of Packets in a Time-Sharing System Using Random Access Communication 8241
Interactive Time-Sharing Command Generating Facility for Conversationally Controlled Tasks 8242
Translator Routine for Querying Data Banks in a Time-Sharing Environment 8248
Programming of Time-Shared Computer Centers for Shipboard Oceanographic Research 8261
Nonlinear Regression Analysis by Time-Sharing Computers 8282
—see also Multiprogramming

Training
Pattern Classification with a Partitioned Training Set 8254
Nonparametric Training Procedure Capable of Producing an Arbitrary Sequence of Patterns 8255
—see also Adaptive, Learning

Translation; Translators (Artificial Languages)
Translator Routine for Querying Data Banks in a Time-Sharing Environment 8248

Trees
File Structures Using Hashing Functions to Define Tree Structures 8234
Properties of the Binary Tree Sort Algorithm 8244
Sampling Approach to Minimal Storage Tree Sorting 8245
Tree Structures for Optimal File Searching and Information Retrieval 8246
IDENTIFIER INDEX

A
Abstract Families of Acceptors 8216
Abstract Families of Languages 8216
Acceleration of Convergence 8255
Acceptance of Sentence Models 8220
Affine Groups 8213
AFL's 8216
ALOHA Time-Sharing System 8241
Alphanumeric Codes 8270
AND-Or Type Subgraphs 8210
Applicability Condition 8215
Architecture (Computer) 8283
Assignment 8210, 8224
Asymptotic Efficiency 8255
Asynchronous Sequential Circuits 8224

B
BCH Codes 8278
Biomedical Applications 8282
Block Codes 8277
Block Floating-Point Arithmetic 8268
Bootstrapping 8236

C
Canonical Forms 8214
Canonical Representation 8213
Carry Propagation 8212
Cascade Canonical Form 8267
Chaining 8246
Channel Coding 8227
Closure 8220
Clustering 8250
Clustering Algorithms 8249
Command Languages 8242
Compatible High Density Bipolar Codes 8271
Complete Deductive Systems 8257
Composite Functions 8212
Conditional Job Control 8242
Conditional Recursive Expressions 8231
Confusion Measure 8250
Congruential Random Number Generators 8263
Connectedness of Networks 8222
Connectivity of Graphs 8247
Consequence Finding 8258
Context of Vertices 8215
Contour Lines 8230
Convergence Acceleration 8255
Convergence of Programs 8231
Correctness of Programs 8231
Correlative Level Coding 8270
CP/MS Time-Sharing System 8235
Craig's Interpolation Theorem 8258
Crostalk in Logic Circuits 8226
Cycles on Networks 8222
Cyclic Algebra 8212
Cyclic Codes 8228
Cyclo-Decomposition 8212
Cyclo-Power Operation 8212
Cyclops System 8256

D
Degenerate Kernels 8262
Deterministic Local Automata 8220
Diagonal Method 8264
Dimensionality Reduction 8250
Diophantine Equations 8259
Directed Graphs 8210, 8247

E
Discrete Systems 8223
Discriminant Plane Logic 8250
Distribution-Free Logic Design 8250
Double Chaining 8246
Dynamic Programming Application 8251

F
Fast Fourier Transforms 8269
Feature Selection 8251, 8252
Feature Space 8249
Fingerprint Matching 8256
Finite Automata 8221
Finite Differences 8261
First-Order Logic 8231
Fletcher–Powell Algorithm 8267
FORTRAN IV Programs 8223
FORTRAN Debugging 8232
FORTRAN Programs 8238, 8259
Fourth Generation Computers 8225
Fredholm Equations 8262
Free Boundaries 8261
Free Surfaces 8261
Fuzzy Sets 8249

G
Galois Fields 8275
Gödel's Compactness Theorem 8258
Gomory's Algorithm 8253
GPSS/360 Applications 8241
Greatest Common Divisors 8259
Green's Functions 8262
Ground Clause 8257
Group Codes 8277

H
Hadamard Matrices 8277
Hamming Codes 8274, 8275
Handprinted Character Recognition 8250
Handwritten Characters 8255
Harmonic Analysis 8213
Hash Addressing 8233
Hash Coding 8233
Hashing Functions 8234
Hidden-Line Problem 8230
High Density Bipolar Codes 8271
Homomorphic Replication 8216
Homomorphisms 8220

I
IBM 360/67 Language 8235
Identification of Functions 8211
Inequality Constraints 8264
Initial-Value Problems 8262
Integers 8259
Interacting Local Automata 8220
Interference (Intersymbol) 8270
Interference of Packets 8241
Interleaved-Bipolar Codes 8271
Interleaved NRZI Recording 8227
Interrupt Handling 8239
Intersection Rule 8221
Intersymbol Interference 8270

J
Jets 8261
JOSTRAN Language 8232

K
Kilburn Adder 8212
Knuth's MIX Computer 8237
Kuno Algorithm 8214

L
Labeled Monadic Algebras 8220
Labeled Graphs 8215
Lagrangian Multipliers 8251
Lamé's Theorem 8259
LAP 6 System 8243
Latin-Square Codes 8273
Layering 8243
Least Squares 8264
Legality of Graphs 8210
Legibility of Displays 8229
Lemke–Spielberg Algorithm 8251
LIN C Applications 8243
Linear Separability 8253
LISP 1.5 Implementation 8235
LISP-Like Expressions 8231
Local Automata 8220
LSI Circuits 8213

M
Mapping 8250
Matching of Fingerprints 8256
Mealy Automata 8219
mix Computer 8237
Mobile Programming System 8236
Modulation Systems 8270, 8271
Monadic Algebras 8220
Monadic Logic 8220
Monte Carlo Method 8260
Monte Carlo Programs 8263
Multimodal Data Sets 8249
Multimodal Methodology 8223
Multivalued Algebra 8212
Multivariable Functions 8260
Multivariate Data Analysis 8250

N
Nondeterministic Finite Automata 8221
Nondeterministic Local Automata 8220
Nonlinear Interpolation 8260
Nonlinear Programming 8264
Nonlinear Regression 8264, 8282
Nonlinear Sequential Networks 8222
Nonlinear Smoothing 8265
Nonlinear Systems 8265
Nonparametric Training 8255
Normal Embedding 8215
Normal Forms 8212
NPS LISP 8235
NRZI Recording 8227
Nuclear Programs 8263
AUTHOR INDEX

A
Abbott, J. L. 8281
Akers, M. N. 8237
Anderson, R. 8257

B
Baer, J. L. 8210
Barzdin, Y. M. 8219
Beus, H. L. 8244
Biswas, N. N. 8211
Bledsoe, W. W. 8257
Bloom, B. H. 8233
Bortels, W. H. 8241
Bossen, D. C. 8273, 8275
Bovet, D. P. 8210
Bradley, G. H. 8259

C
Callender, E. D. 8225
Chandy, K. M. 8238
Chien, R. T. 8273
Chow, J. C.-M. 8254
Coffman, Jr., E. G. 8234
Crampton, T. H. M. 8263
Croisier, A. 8271
Cross, B. 8280
Cryer, C. W. 8261

D
DeFalco, J. A. 8226
Desens, R. B. 8230

E
Estrin, G. 8210
Eve, J. 8234

F
Franszek, P. A. 8272
Frazer, W. D. 8245
Fricks, C. L. 8240
Fu, K. S. 8252

G
Gabrielian, A. 8220
Gentry, D. G. 8235
Ginsburg, S. 8216
Gitman, I. 8249
Gold, B. 8266
Graham, W. R. 8232
Grant, C. A. 8242
Guinn, D. F. 8269

H
Heidt, J. S. 8240
Homer, L. D. 8282
Horwitz, D. L. 8282
Hsiao, M. Y. 8273, 8274

I
Ibaraki, T. 8253
Ichida, K. 8260

K
Kagiwada, H. H. 8262
Kalaba, R. 8262
Kameda, T. 8221
Klappholz, D. 8248
Klir, G. J. 8233
Klososky, R. A. 8283
Knast, R. 8217
Kobayashi, H. 8227, 8270

L
Lechner, R. J. 8213
Leondes, C. T. 8265
Levine, M. D. 8249
Levy, D. M. 8251
Li, T. J. 8252
Lum, V. Y. 8228

M
MacNeilage, D. C. 8232
Maki, G. K. 8224
Manna, Z. 8231
Marill, T. 8256
Marin, M. A. 8223
McGonegal, C. A. 8266

N
Nelson, G. D. 8251

O
Oppenheim, A. V. 8268

P
Patel, A. M. 8277
Peller, J. B. 8265
Pickholtz, R. L. 8278
Pneuli, A. 8231

R
Rabiner, L. R. 8266
Ramamoorthy, C. V. 8238
Rocher, E. Y. 8278

S
Sammon, Jr., J. W. 8250
Savage, J. E. 8276
Shaw, A. C. 8247
Shrager, R. I. 8264
Slagle, J. R. 8258
Smith, W. D. 8239
Spanier, E. H. 8216
Stanfel, L. E. 8246
Stear, E. B. 8265
Steiglitz, K. 8267

T
Tang, D. T. 8227, 8228
Tracey, J. H. 8224
Tsuda, T. 8260
Turner, E. 8256

V
Vartebedian, A. G. 8229

W
Waite, W. M. 8236
Wang, T.-Y. 8239
Weiner, P. 8221
Whelchel, Jr., J. E. 8269
Wilkes, M. A. 8243
Woods, W. A. 8214

Y
Yajima, S. 8218
Yasui, T. 8218
Yates, J. 8256