Abstracts of Current Computer Literature

These abstracts and the associated subject and author indexes were prepared on a commercial basis under the direction of Dr. Geoffrey Knight, Jr., Consultant to Cambridge Communications Corp., 1612 K Street, Washington, D.C., who publish Information Processing Journal, Solid State Abstracts Journal, Electronics Abstracts Journal, Cumulative Computer Abstracts, Cumulative Solid State Abstracts, and Cumulative Electronics Abstracts.

CONTENTS

ABSTRACTS .......................................................... Pages 609-617
0) GENERAL; PEOPLE AND SOURCES; EDUCATION .... 5705-5707
1) LOGIC AND SWITCHING THEORY; SEQUENTIAL MACHINES ........................................ 5708-5714
2) DIGITAL COMPUTERS AND SYSTEMS .................. 5715-5721
3) LOGIC DEVICES AND CIRCUITS (HARDWARE) .... 5722-5723
4) DIGITAL STORAGE AND INPUT-OUTPUT EQUIPMENT .............................................. 5724-5731
5) PROGRAMMING AND CODING OF DIGITAL MACHINES ............................................... 5732-5747
6) HUMAN COMMUNICATION, DOCUMENTATION, AND HUMANITIES .................................... 5748-5749
7) BEHAVIORAL SCIENCE, PATTERN RECOGNITION, AND ARTIFICIAL INTELLIGENCE ............. 5750-5755
8) MATHEMATICS .................................................... 5756-5762
9) PROBABILITY, MATHEMATICAL PROGRAMMING, DIGITAL SIMULATION, INFORMATION THEORY, AND COMMUNICATION SYSTEMS .............................. 5763-5766
10) SCIENCE, ENGINEERING, AND MEDICINE ............ 5767
11) ANALOG AND HYBRID COMPUTERS ................. —
12) REAL-TIME SYSTEMS AND AUTOMATIC CONTROL; INDUSTRIAL APPLICATIONS .............. 5768
13) GOVERNMENT, MILITARY, AND TRANSPORTATION APPLICATIONS ........................... —
14) BUSINESS APPLICATIONS OF INFORMATION PROCESSING ......................................... 5769-5770

SUBJECT INDEX ....................................................... Pages 617-620
AUTHOR INDEX ....................................................... Page 620
0) GENERAL; PEOPLE AND SOURCES; EDUCATION


The report is divided into two main parts. The first part is a tutorial discussion of present developments in the computer field, both for hardware and for software. The second part discusses potential future applications for the computer.


The vital role that electrical engineering departments must play in providing undergraduates with special competence in computer sciences is explored. Three related problem areas are discussed: 1) meeting the needs of students majoring in computer sciences in electrical engineering; 2) balancing the treatment of continuous and discrete systems; and 3) realizing wider and more effective use of the digital computer as a tool for analysis and design in all engineering courses. Specific suggestions for meeting these needs are offered.


A procedure to supply test data for a number of undergraduate programming exercises in the PL/1 language and to check the validity of programs is described. The procedure provides diagnostic information to the student and performs all necessary output, as well as maintaining complete records of student performance on magnetic disc storage. The procedure differs from many previous grading routines in being called as a precompiled library subroutinel and is the first known grading procedure for PL/1. The initial set of class problems and specimen output listings are appended.

1) LOGIC AND SWITCHING THEORY; SEQUENTIAL MACHINES

Completeness Results in the Mathematical Theory of Computation—see 5756.


Functionally packaged logic can only be effectively utilized if the totality of switching functions that each package is capable of providing is recognized. Theorems concerning, and algorithms operating on, multiple output switching functions (possibly with don't care conditions) in cubical array notation are presented that 1) detect partial symmetry and redundancy sets of input variables, 2) determine the function generated by a package with some of its inputs tied to logical 1 or 0 or tied together, and 3) rapidly show equivalence between two functions using symmetry information. While manual execution of the algorithms is possible, they are computer oriented. Results from actual computer experimentation show their efficiency.


The hyperoctahedral group on the set of N-variable Boolean functions is defined. The problem of determining, given an arbitrary function, the class membership (type) is posed and, for the special case of four-variable functions, solved by a method that in use proved to be four times as rapid as computation of the conventional Golomb–Ninomiya invariants.


In this note the successive-higher-ordering method for testing and realization of threshold functions is applied to the realization of a threshold function F, such that it will contain a given function Fp, which may or may not be a threshold function, and such that it will be contained in if Fp is the function representing the don't care vertices. Before the application of successive-higher-ordering method, the given functions and are first changed into unate functions by the successitive positivizing of the functions with respect to the variables, one at a time. Some theorems relating to this formation of unate functions are presented. A systematic procedure for testing and realization is developed. An example is given for illustration.


A synthesis procedure is introduced for obtaining multithreshold threshold-element realizations of arbitrary Boolean functions including functions with don't cares. Through transforms of multithreshold threshold-element realizations, the procedure is also applicable to threshold-element-network realizations of functions. The procedure is useful for hand calculations for functions with a small number of variables, and has been programmed on a computer for functions having up to 11 variables (the number of variables is presently limited by storage allocation). The function to be realized is first completely decomposed about its variables to form a tree. The functions resulting at the tips of the tree branches form the complex sum of simple "value functions" (which specify a set of multithreshold realizations) that are uniquely determined by the weight chosen for the final variable in the decomposition. After choosing this weight, the given function is systematically reconstructed from the tree by choosing the weight of one variable at a time through the combined use of a theorem that specifies the allowable weights of the new variable and an analysis of the minimum possible number of thresholds required as a result of the given allowable weight. Experimental results of hand and computer calculations show the procedure to be fast in producing realizations with a small number of thresholds and a small sum of weight magnitudes.

Threshold Logic Network for Recognizing Shape Invariance—see 5752.


Arbitrary finite automata are decomposed into their major substructures, the primaries. Several characterizations of homomorphisms, endomorphisms, isomorphisms, and automorphisms of arbitrary finite automata are presented via reduction to the primaries of the autamata. Various characterizations of these transition-preserving functions on singly generated automata are presented and are used as a basis for the reduction. Estimates on the number of functions of each type are given.


The purpose of this paper is to describe an algorithmic “solution” to the assignment problem of synchronous sequential machines. The figure of merit used provides a mathematical evaluation of the reduced dependencies that may exist in the set of logic equations. If desired, the algorithm can assign the input, state, and output symbols of a given machine so as to "minimize" the total logic; i.e., reduced dependencies of both the state and output logic on state and input variables are optimized. The method is nonenumerative in the sense that the first assignment found is optimal. A restricted version of the algorithm has been programmed for an IBM 7094 computer.


In this paper the construction of a switching network capable of n-permuta-

tion of its input terminals to its output terminals is described. The building blocks for this network are binary cells capable of
ABSTRACTS OF CURRENT COMPUTER LITERATURE

permuting their two input terminals to their two output terminals. The number of cells used by the network is \((n \log_2 n - n + 1) = \sum \log_2 n - \log_2 k\). It could be argued that for such a network to have this number of cells is a lower bound, by noting that binary decision trees in the network can resolve individual terminal assignments only and not the partitioning of the permutation set itself which requires only \((\log_2 n) = \sum \log_2 n - \log_2 k\) binary decisions. An algorithm is also given for the setting of the binary cells in the network according to any specified permutation.

2) DIGITAL COMPUTERS AND SYSTEMS


Binary multiplication can be speeded up by taking two or more bits of the multiplier at a time. This note describes an exact method, based on the use of a discrete-time, finite-state system, for calculating the gain in multiplication speed resulting from such a scheme. It is shown that the gain in speed is less than what had previously been obtained using an approximation formula.


Data transfers in computing systems with memory hierarchies usually prolong computing time and, consequently, cause degradation of system performance. A method to determine data processing rates and the relative utilization of memories for various system configurations under a variety of program loads is presented. According to this method, a program-independent ultimate data processing rate is derived from characteristics of the processor and the fast-access memory of the system. The degradation factors are determined by combining statistics of the data flow of actual programs and hardware parameters of the processor and all memories. The statistics of data flow in the memory hierarchy are obtained by analyzing a number of recorded address traces of executed programs. The method presented permits quick evaluation of system performance for arbitrary time periods and for maximum and minimum concurrence of operation of processors and memories.

CDL 1 Language for Describing Computer Systems—see 5739.


This paper treats the problem of automatic fault diagnosis for systems with multiple faults. The system is decomposed into \(n\) units \(u_1, u_2, \ldots, u_n\), where a unit is a well-identifiable portion of the system which cannot be further decomposed for the purpose of diagnosis. By means of a given arrangement of testing links (connection assignment) each unit of the system tests a subset of units, and a proper diagnosis can be arrived at for any diagnosable fault pattern. Methods for optimal assignments are given for instantaneous and sequential diagnosis procedures.


As digital system speeds increase and their sizes diminish, it becomes increasingly important to understand the mechanism of signal crosstalk (noise) in interconnections between logic elements. The worst case is when two wires run parallel for a long distance. Past literature has been unsuccessful in explaining crosstalk between parallel wires above a ground plane, because it was assumed that only one signal propagation velocity was involved. This paper proves that a signal introduced at one end of a printed wire above a ground plane in the presence of a second parallel (passive) wire must break up into two signals traveling at different velocities. The serious crosstalk implications are examined. The new terms slow crosstalk (SN), fast crosstalk (FX) and differential crosstalk (DX) are defined.

Low-Level Data Structure Package for List Processing on the FD7P Computer—see 5732.

3) LOGIC DEVICES AND CIRCUITS (HARDWARE)


Two circuit models of a semiconductor junction rectifier are presented. For times long compared to the effective minority carrier lifetime, the usual current rectifier concept obtains. For times short compared to the lifetime, a charge rectifier is better able to explain experimental results; reverse recovery is evidence of this. Viewed in this way, semiconductor junctions are suitable for certain types of digital information processing. Voltage gain is limited only by junction breakdown and driver capability, but charge (current) gain is inevitably less than one. The amount of charge run-down is shown to depend on the time the information is in the circuit. Scan generators, shift registers, serial memories, and serial converters have been studied as examples of the wide range of applications for which these devices are suitable.


To date there have been no direct measurements of the switching speed of an individual crossed-film cryotron (CFC) due to the extremely low gate resistance of the device in its normal state. A method which is, in principle, similar to conventional sampling techniques is used to determine the CFC switching speed with a 2-ns time resolution and a gate resistance sensitivity of 0.01 ohm. CFC switching speeds are determined as a function of control current overdrive and gate current. In this way, the
gain-bandwidth limitations of the device are experimentally determined. These data can be used to determine the optimum speeds of CFC logic circuits.

4) DIGITAL STORAGE AND INPUT-OUTPUT EQUIPMENT

Performance Evaluation of Computing Systems with Memory Hierarchies—see 5716.

Simulation of a Multiprocessor Using Associative Memories for Executive Control—see 5721.

Serial Memories Using Charge Rectifiers—see 5722.


A cross-sectional model is designed to prove the feasibility of a 147-kbit thin magnetic film memory in the eddy current braked nondestructive readout (NDRD) mode. With the full word, digit, and sense line lengths, the model is operated with drive currents smaller than 400 mA. The write and the nondestructive read word pulses are different in amplitude and duration, but fit the requirements of a high-speed selection matrix. Including the matrix, a nondestructive read cycle time of 20 ns is achieved, yielding sense signals in the order of 1 nV. Word and digit noise compensation schemes are investigated at large duty factors. Bipolar digit pulses are used.


Planar processing techniques for producing coupled film storage devices are presented. The discussion centers around a process sequence for a coupled easy-axis structure built on a metal substrate, largely by vacuum deposition techniques. The process results in packing density of about 8000 bits/in². Included is a test of one such element displaying useful memory device characteristics under conditions of a worst-case pulse test program.


A magneto-optic memory element which was first proposed by Chang et al. has been investigated. The element is based upon the properties of magneto-optic materials with a compensation temperature. It is a single crystal wafer of gadolinium iron garnet. The importance of domain walls in limiting the minimum size of a bit was recognized. This limitation was removed by scribing the wafer into square bits approximately 25 microns on a side. The reading operation was improved by utilizing a beam splitting procedure similar to one suggested by Miyata and Lentz. The optimum thickness of the memory element and the optimum wavelength for memory operation were determined. Thermal considerations were investigated. The laser power required for the writing operation and limits for the writing speed set by thermal factors were determined. Experimental results were in good agreement with these theoretical predictions.


A magnetic-film memory accessed by combined photon and electron beams is proposed. The electron beam is used to heat a selected bit, which results in lowering the switching threshold so that information can be written selectively into that bit by means of an external magnetic field. Reading is accomplished by simultaneously illuminating a bit with a laser and a photon beam. Then a thermally modulated magnetic-optical signal is generated by intensity modulation of the electron beam. This arrangement is advantageous since a high-resolution photon beam and photon deflector are not required. The frequency response for thermally modulating a 1-µm bit is calculated to be ~500 MHz; the necessary temperature dependence of the magneto-optical coefficient and the coercive force can be obtained by using composite films made from layers having different Curie points. Various magneto-optical configurations are readily devised which yield the value of a bit (one or zero) in terms of the phase (zero or 2π) between the magneto-optical signal and the electron-beam intensity modulation. The shot noise limited signal-to-noise ratio (SNR) is determined by heating of a bit from the photon beam. It is calculated that for a low-loss magneto-optical material such as EuO, a 1-µm bit can be read in 1 µs when illuminated with a 1000-µm photon beam. The base-line temperature rise due to heating from the photon beam can be kept small by using narrow pulses of light (width ~10⁻⁶ second), as, for example, from a mode-locked laser.


A new type of electron beam activated switch (EBAS) which describes utilizes electron beam induced charge storage in the metal-oxide-semiconductor (MOS) system. The state of the EBAS is determined by monitoring the surface conductance of the semiconductor. After discussing the basic charge-storage phenomena, memory arrays that use the electron beam for storing and reading information are described. A matrix array of EBASs in which information is stored using the electron beam and read by row-column access circuits is discussed in detail. The time to store a bit of information is a function of the current density of the electron beam; an approximate dosage of 10⁻⁹ C/cm² is required for storage. A memory design using Schlesinger’s microspot tube for the electron optics is discussed. It is shown that storage of 1.0X10¹⁰ bits per tube should be possible with presently available electron-optical design and semiconductor technology.

Memory Circuits Using a Simulated Neuron—see 5754.


Closed form expressions have been obtained for the power density spectra of signal waveforms in use for digital magnetic recording. The “signal” is the magnetization versus distance profile imposed on the magnetic medium to encode binary ones and zeros. Two general classes of signals are distinguished in terms of neighboring bit-to-bit correlation. Standard recording methods such as saturation NRZ and phase modulation are considered, as well as techniques requiring bias currents, such as sinusoidal frequency-shift modulation. The spectra are compared with frequency response curves for the reading head at several recorded bit densities. The more ideal bandwidth characteristics of the newer phase- and frequency-modulation techniques help to explain the improved performance at high bit densities.


A low-cost remote terminal which provides output in switch form from a time-shared digital computer is described. The terminal consists of a modified model 35 KSR teletype and a local memory unit. The unit is independent of the particular computer, and is easy to test and maintain. The states of the memory control and memory words are observable directly by indicator lights. An application of the memory to the automatic set-up and control of an analog computer are displayed on an oscilloscope; this makes possible, for example, the rapid display of time response of linear systems, under digital program control.

Mechanism for Communicating with a Computer in Natural English—see 5748.

Interactive Displays for Document Retrieval—see 5749.

Man-Machine Communication for On-Line Classification of Chinese Characters—see 5751.

The paper describes a proposed microprogrammed interface computer, "Intergraphic," which will link many (initially 13, potentially more than 50) general-purpose graphical terminals to a central processor. Intergraphic will generate new images once only, at high speed (10-MHz incremental plotting rates) on one of several small, electrostatically deflected, precision CRT's. The images will be generated on a 1024 by 1024 grid in incremental and random-point display modes by fast microcode sequences which interpret display lists from the central processor. The centrally generated images will be scan converted to standard television video signals and recorded on a multitrack video disk(s), each track refreshing a low-cost standard television terminal. User feedback will be via raster coordinates, determined by a light-pen ("raster-pen") and simple counting circuits at each terminal. The paper concentrates on the digital interface structure which is versatile and fast (3-5-ns integrated circuits and 100-ns cycle-time read-only memory). The description centers on display generation, although the structure is largely general purpose. The proposed operating system is outlined only since it is in an early stage of development. Also, the order-code accompanying data from the central processor is incomplete. However, new orders can be readily interpreted by adding microprograms to the read-only memory (i.e., "firmware," not hardware extensions); thus, the interface will also be a useful medium for experimentation in graphical structures and communication.

Solution of the Two-Dimensional Hidden-Line Problem—see 5767.

5) PROGRAMMING AND CODING OF DIGITAL MACHINES

5732


A low-level data structure package for the PDP7 computer is described. Its principal features are the compact form in which given structures may be set up and the wide range of formats permitted. Ring structures are regarded as special cases of general list structures, and the package permits the generation and processing of all legal list structures. It is, however, specifically oriented to a certain class of uni-directional list and ring formats, and achieves particularly good space utilization when they are used. Space statistics for the package are presented which, when compared with the performance of more conventional schemes, show typical savings of about 30 percent.

5733


Comparing properties of non- and self-modifying programs leads to the definition of independent and dependent instructions. Non-modifying programs contain only indepen-endent instructions, and such programs can be analyzed by a straight-forward, two-step analysis: explicitly, the program control flow is detected; second, that control flow is used to determine program data flow or data processing. However, self-modifying programs can also contain dependent instructions, and then program control flows and data processing are not necessarily independent. This cyclic interaction suggests using an iterative or relaxation analysis technique.

Initially, the relaxation procedure determines a first approximation to control flow, the second step, to data flow. These two steps are repeated until steady-state conditions are reached. Algorithms for implementing the first iteration are presented. These algorithms are capable of analyzing programs which modify their control and processing instructions while executing. Also described are data structures which permit constructing functional expressions for data flow or information processing. Finally, actual output flowcharts of self-modifying programs are displayed.

Microprogrammed Graphical-Interface Computer—see 5731.

5734


Recent developments of programming languages have led to the emergence of language features which introduce or exacerbate certain problems: the proliferation of new elements defy every control exercised by the designers, and the nature of the new cells often proved to be incompatible with the existing body. In order that a language be free from such symptoms, it is necessary that it be built upon basic concepts which are sound and mutually independent. The rules governing the language must be simple, generally applicable, and consistent. In order for simplicity and consistency to be achieved, the fundamental concepts of a language must be well-chosen and defined with utmost clarity. In practice, it turns out that there exists an optimum in the number of basic concepts, below which not only implementability of these concepts on actual computers, but also their appeal to human intuition becomes questionable because of their high degree of generalization. The information notes presented do not abound with ready-made solutions, but it is hoped they shed some light on several related subjects and inherent difficulties. They are intended to summarize and interrelate various ideas which are partly present in existing languages, partly debated within the IFIP Working Group 2.1, and partly new.

5735


A method for defining programming languages is developed which introduces a rigorous relationship between structure and meaning. The structure of a language is defined by a phrase structure syntax, the meaning in terms of its interpretation, the execution of a sequence of interpretation rules exerts upon a fixed set of variables, called the environment. There exists a one-to-one correspondence between syntactic rules and interpretation rules, and the sequence of executed interpretation rules is determined by the sequence of corresponding syntactic reductions which constitute a parse. The individual interpretation rules are explained in terms of an elementary and obvious algorithmic notion. A constructive method for evaluating a text is provided and for certain decidable classes of languages their unambiguity is proven. As an example, a generalization of ALGOL is described in full detail to demonstrate that concepts like block-structure, procedures, parameters, etc., can be defined adequately and precisely by this method.

5736


A programming language for the IBM 360 computers and aspects of its implementation are described. The language, called PL360, provides the facilities of a symbolic machine language, but displays a structure defined by a recursive syntax. PL360 was designed to improve the readability of programs which must take into account specific characteristics and limitations of a particular computer. It represents an attempt to further the state of the art of programming by encouraging and even forcing the programmer to improve his style of exposition and his principles and discipline in program organization. Because of its inherent simplicity, the language is particularly well suited for tutorial purposes. The attempt to present a computer as a systematically organized entity is also hoped to be of interest to designers of future computers.

Grading Subroutine for Student Programming Exercises in the PL/I Language—see 5707.

5737


This paper describes the development of an ALGOL program to perform syntax analysis, using a set of syntax definitions supplied in a notation similar to Backus Normal Form. Five applications of syntax analysis are then discussed, including algebraic differentiation and compilation.

5738


DASH (Dynamic ALGOL String Handling) is a set of procedures designed to extend ALGOL to the expression of non-numerical or partly non-numerical algorithms for which it is normally unsuited.

The objective of this report is to develop a formal language to describe hardware and software computing systems. The language is to provide a linguistic basis to consider machine-aided solutions of a variety of design problems; i.e., problems concerning design documentation, data retrieval systems, system simulation, diagnosis, analysis and synthesis. This report discusses in some detail the considerations that went into the design of the computer description language, called CDLI; it points out the need for developing such a language and briefly discusses the kinds of applications such a language may have. The report points out the various aspects of system description one may encounter in a design process and relates them to the language features necessary to express them; the language itself is described informally. Examples are presented to illustrate the use of the language, the concepts associated with descriptions of systems at various stages of design, and the consequent hierarchical structure such descriptions acquire.

Translator Writing Systems, J. Feldman and D. Gries (Stanford U.); Commun. ACM, vol. 11, pp. 77–113, February 1968. A critical review of recent efforts to automate the writing of translators of programming languages is presented. The formal study of syntax and its application to translator writing, various approaches to automating the postsyntaxic (semantic) aspects of translator writing, and several related topics are covered.


The construction of efficient parsing algorithms for programming languages has been the subject of many papers in the last few years. Techniques for efficient parsing and algorithms which generate the parser from a grammar or phrase structure system have been derived. Some of the well-known methods are the precedence techniques of Floyd, and Wirth and Weber, and the production language of Feldman. Perhaps the first such discussion was by Samelson and Bauer. There the concept of the push-down stack was introduced, along with the idea of a transition matrix. A transition matrix is just a switching table which lets one determine from the top element of the stack (denoting a row of the table) and the next symbol of the program to be processed (represented by a column of the table) exactly what should be done. Either a reduction is made in the stack, or the incoming symbol is pushed onto the stack. Considering its efficiency, the transition matrix technique does not seem to have achieved much attention, probably because it was not sufficiently well-defined. The purpose of this paper is to define the concept more formally, to illustrate that the technique is very efficient, and to describe an algorithm which generates a transition matrix from a suitable grammar. The report also describes other uses of transition matrices besides the usual ones of syntax checking and compiling.


The continuing controversy over the relative merits of time-sharing versus batch processing has taken a new and significant turn from predisciplinary speculation to applied scientific experimentation. Within the last two years, five experimental studies have appeared in the literature, each comparing some form of online and offline data processing with respect to man-machine measures of system performance. These five pioneering studies comprise the first substantive database for comparing and evaluating experimental methodology and findings bearing on the growing and changing competition between time-sharing and batch processing systems. This paper provides a critical review of these five experiments, summarizes findings, problems and pitfalls, and offers recommendations for future experimental work.


The document presents a user's description of the TRACE system, which provides an on-line technique for scanning data and deriving variables. It is divided into two main sections: the first a tutorial guide introducing the user to the basic principles of the system, and the second a reference guide to the entire body of the TRACE program. The user is shown how to initiate an interaction with the time-sharing system, how to employ every capability of TRACE, what errors may be expected in operation, and what statistical products may be derived through use of the program. A complete index allows the user to refer readily to any portion of the document.

Low-Cost Terminal for Time-Shared Computers—see 5730.


The report contains features of executive, operating or monitor systems considered important for evaluation and comparative analysis. These features are identified in a form expressly for inclusion in the three step method for software evaluation: executive, operating, or monitor systems. Included in this paper is a composite list of functions contained in current executive systems. These functions provide the basis for a standard approach to the software aspects of executive systems particularly needed for evaluation and comparative analysis.


Scheduling and allocation of computing facilities is generally carried out in a very primitive and ineffective way. The difficulty stems from a number of sources including: 1) semantic confusion, 2) failure to consider all relevant factors, and 3) failure to establish a global context for analysis of specific operational problems. This paper attempts to establish the required context by considering all relevant factors and by carefully sorting out the semantics. Specifically, market principles are employed to characterize the environment at its most general level. At the detailed level, a model based on the dimensions of 1) demand magnitude, 2) interaction rate, and 3) deadline penalty severity is presented.


The report attempts to answer some of the questions concerning the advantages and disadvantages of time-sharing. To accomplish this, the general problem of evaluating computer system performance is first addressed. General system characteristics are specified that include the computer and its operating system, and users and their jobs. The main emphasis is placed upon the operating system; the effects of having different computers, users, or jobs are treated as parameters. The most important evaluation criterion is considered to be cost, which includes both user cost and computer system cost. Quantitative models are developed that describe computer center users, the programs they run, and the different operating systems they might use.


The general features of program overlay systems are described. Three main types—automatic, semiautomatic, and nonauto-
ABSTRACTS


Although a great deal of effort has gone into programs that parse English sentences into representations of their syntactic structure, a facility for actually communicating with the machine in English has not been realized because of the inability of the machine to interpret the "meaning" of a sentence. This paper describes a proposal for translating a representation of the syntactic structure of a sentence into an expression in a formal language that represents the "meaning" of the sentence to the computer. This facility in conjunction with a facility to parse English sentences would provide the necessary mechanism for communicating with the computer in natural English. The specific problems with which this paper deals is that of using the computer to answer English questions.

Implementation of Syntax Analysis Using ALGOL—see 5737.


Interactive computer systems establish a dialog between the man and the machine. By means of displays, the user receives immediate feedback of the results of his actions, and he is able to modify his decisions in order to obtain a system response that is most relevant to his needs. BOLD (Bibliographic On-Line Display) is an example of a highly automated interactive document storage and retrieval system that is in operation at System Development Corporation. It enables the user to browse through the data base by subject categories or search for specific documents. The response is rapid and the results are satisfying.

7) BEHAVIORAL SCIENCE, PATTERN RECOGNITION, AND ARTIFICIAL INTELLIGENCE


This paper presents the dynamic programming approach to the design of optimal pattern recognition systems when the costs of feature measurements describing the pattern samples are of considerable importance. A multistage or sequential pattern classifier which requires, on the average, a substantially smaller number of feature measurements than that required by an equally reliable nonsequential classifier is defined and constructed through the method of recursive optimization. Two methods of reducing the dimensionality in computation are presented for the cases where the observed feature measurements are 1) statistically independent, and 2) Markov dependent. Both models, in general, provide a ready solution to the optimal sequential classification problem. A generalization in the design of optimal classifiers capable of sequentially selecting a sequence of feature measurements is also discussed. Computer simulated experiments in character recognition are shown to illustrate the feasibility of this approach.


This paper describes a method for using sequential positional information to recognize handprinted Chinese characters, and a computer program which uses this method to provide a translation aid. The method is based on the fact that Chinese characters are drawn as a definite sequence of a few types of strokes. The program, which provides a direct man-computer communication via a RAND Tablet and a CRT display, simplifies the process by reducing the number of steps and the time required.


A property filter is developed that is suitable for recognizing translation-rotation-dilation classes of two-dimensional images. Invariant outputs corresponding to such classes are obtained by employing two successive sampled spatial harmonic transforms. The required analyses are equivalent to taking inner products of pairs of vectors only one of which is variable in each case. Subsequently, the necessary network may be realized with fixed threshold logic, independent of the character classes to be recognized. The effectiveness of the property filter has been confirmed with printed and handwritten numerals by coupling it to a standard adaptive categorizer of a type assuming linear separability. There is further evidence to show that performance is improved by coupling a categorizer that does not assume linear separability.


An unsupervised or non-supervised adaptive algorithm for linear decision boundaries is applied to two pattern recognition problems: the classification of spoken words, and the classification of hand-printed characters. The term unsupervised indicates that the class identification of the input patterns is not continuously available to the adaptive system. The algorithm discussed offers two advantages for pattern recognition applications. First, the number of patterns which must be labeled with class identification is reduced. Second, the adaptive system can change in class distribution over time, due to data fluctuation or hardware degradation. These advantages are demonstrated for each of the two applications.


The Simuron (simulated neuron) is a distributed active element analogous to the neuron and has learning and oblivious mechanisms. Like the neuristor, it can be coupled by two types of junctions. This letter proposes frequency memory and logical memory circuits using the Simuron.


The heuristic program discussed searched for a constructive proof or disproof of a given proposition. It uses a search procedure which efficiently selects the seemingly best proposition to work on next. This program is a multipurpose in the sense that the types of problems it can handle are varied. As an initial experiment, the program was given the task of searching for proofs and disproofs of propositions about a simple game. Kalah is a two-person game. In another experiment the program, after some modifications, played the game of kalah. This program was compared with another tree-searching procedure, the Alpha-
8) MATHEMATICS


This paper presents the possibility of improving Birstow’s algorithm for finding the roots of polynomials with real coefficients. It discusses points concerning the removal of factors of any chosen order and with speeding the convergence rate. The case of multiple factors is also examined.


It is shown how many previous methods for the exact solution (or best least-squares solution) of systems of nonlinear equations are all based upon simple cases of the generalized inverse of the matrix of first derivatives of the equations. The general case is given and algorithms for its application are suggested, especially in the case where the matrix of first derivatives cannot be calculated. Numerical tests confirm that these algorithms extend the range of practical problems which can be solved.

Application of Syntax Analysis to Algebraic Differentiation—see 5737.


This paper deals with integrals where the integrand contains a weight function which becomes infinite in one or both ends of the integration interval. An evaluation technique based on Romberg’s method is described, and several examples are discussed.


A method for the numerical solution of a Fredholm integral equation of the first kind is derived and illustrated. The method employs an a priori constraint vector together with covariances of both the constraint vector and the measurement errors. The method automatically incorporates an optimum amount of smoothing in the sense of maximum-likelihood estimation. The problem of obtaining optimum basis vectors is discussed. The trace of the covariance matrix of the error in the solution is used to estimate the accuracy of the results. This trace is used to derive a quality criterion for a set of measurements and a given set of constraint statistics. Examples are given in which the behavior of the solution as obtained from a specific integral equation is studied by the use of random input errors to simulate measurement errors and statistical sampling. The variation and behavior of the trace of the error covariance matrix for various bases is also illustrated for the equation being considered.


In this paper the problem of readily determining the inverse Laplace transform numerically by a method which meets the efficiency requirements of automatic digital computation is discussed. Because the result is an inverse function is given as a Fourier cosine series, the procedure requires only about ten FORTAN statements. Furthermore, it does not require the use of involved algorithms for the generation of any special functions, but uses only cosines and exponentials. The basis of the method hinges on the fact that in evaluating the inverse Laplace transform there exists a freedom in choosing the contour of integration. Given restrictions, the contour may be any vertical line in the right-half plane. Specifying a line, the integral to be evaluated is essentially a Fourier integral. However, the method is concerned with determining the proper line, so that when the integral (along this line) is approximated, the error is as small as desired by virtue of having chosen the particular contour.

9) PROBABILITY, MATHEMATICAL PROGRAMMING, DIGITAL SIMULATION, INFORMATION THEORY, AND COMMUNICATION SYSTEMS


Pseudorandom number generator programs can be modified by a simple procedure which has particular advantages for the shorter word binary computers. The method combines the result of any conventional arithmetic generator with the current state of counters by means of a logic instruction. A variety of statistical tests have indicated that good results are realized.


An approximate but fairly rapid method for solving linear integer programming problems is presented, which utilizes, in part, of some of the philosophy of “direct search” methods. The method is divided into phases which can be applied in order and has the desirable characteristic that a best feasible solution is always available. Numerical results are presented for a number of test problems. Some possible extensions and improvements are also presented.


The present paper describes a simple method for finding an indicator of the error in solutions found by a procedure using the simplex-technique, and also describes a method for improving these solutions to any desired accuracy provided certain conditions are met.

Dynamic Programming Approach to Sequential Pattern Recognition—see 5750.


Self-clocking of binary information involves a coding constraint whereby the maximum length of zero strings is limited to a predetermined number. For reasons of operational stability it is desirable to keep this number low. The present system is a degradation in code density. The class of serial recording which includes the well-known frequency modulation and phase encoding represents one extreme where every other symbol is mandatory, making the code density one bit for every two symbols. This paper describes a coding method which improves code efficiency while main-
taining the constraint on the length of zero strings. It will be shown that adaptive coding, which is an extension of the block code concept known from information theory, offers a significant improvement. The concept is illustrated by describing a practical application in which the code density is improved from two symbols per bit to three symbols per bit pair, while keeping the maximum length of zero strings to one. It is shown that while this technique represents an improvement by one third over phase encoding, it approaches the theoretical limit for this class of codes to within 4 percent.

10) SCIENCE, ENGINEERING, AND MEDICINE

5767

The two-dimensional "hidden-line" problem is the problem of determining, by means of a computer algorithm, which edges or parts of edges of an arbitrary, nonintersecting polygon are visible from a specified viewpoint on the plane of the polygon. The problem is an important one in the field of computer graphics, and is encountered, for example, in using a computer to determine the portion of an island's coastline visible from a ship offshore. Some propositions are introduced that facilitate the solution of this problem. A general algorithm for the solution is described, and illustrative examples are given of hidden-line problems solved with a digital computer.

11) ANALOG AND HYBRID COMPUTERS

Use of Hybrid Computing in the Analysis of Steel Rolling—see 5770.

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

5768

In formalizing the control and continuous industrial regulation algorithms by means of the table-address method and by generalized programming languages, various researchers established the fact that the given algorithm consists of a series of periodically repeating standard operators. It became obvious that these operators should be realized within the system of commands of control computers. The present authors, consequently, present an appropriate system of commands and explain the writing down of such a system. They divide the control and the regulation algorithm into separate operations in such a way that the algorithm is divided into a minimum number of operators with a maximal repetition rate.

14) BUSINESS APPLICATIONS OF INFORMATION PROCESSING

5769

In commercial and administrative ADP it often happens that there is one large master file and many output files are to be made from this. One possible solution to avoid using the master file many times is to construct from it a few extraction files and from these compile the output files. In this paper a method is described to define the extraction files and to optimize their contents by zero-one programming. An algorithm and an example are presented in a simplified case.

5770

The equations that govern the behavior of steel undergoing reduction are complex differential equations. A method of solving these equations using a hybrid computer is described, and the peculiar problems associated with such a solution are discussed. In the hybrid implementation, differential equations can be solved very fast, and the speed of solution is governed by the number of samples one requires of the trajectory from the analog computer for useful computation in the digital computer. Thus, methods are described whereby the number of samples taken can be reduced and the accuracy of the manipulations performed by the digital computer can still be maintained. The results obtained are both accurate and rapid, and the system has the normal engineering appeal of a purely analog solution.

SUBJECT INDEX

A
Adaptive
Nonsupervised Adaptive Algorithm for Linear Decision Boundaries in Pattern Recognition 5753
Adaptive Coding for Self-Clocking Recording 5766
—see also Control, Patterns
Algebra; Algebraic
Application of Syntax Analysis to Algebraic Differentiation 5737
Solution of Systems of Simultaneous Linear Equations by Elimination 5758
Least-Squares Solution of Systems of Non-Linear Equations by Inverse Methods 5759
—see also Matrices, Polynomials
Algorithms
Construction of Efficient Parsing Algorithms for Programming Languages 5741

TRACE Time-Shared Routines for Analysis, Classification and Evaluation of Statistical Data 5743
Generalization of Bairstow's Algorithm for Finding Roots of Polynomials 5757
Algorithms for:
Identification of Symmetry, Redundancy and Equivalence of Boolean Functions 5708
Assignment Problem of Synchronous Sequential Machines 5713
Construction of a Permutation Switching Network 5714
Comparison of Properties of Non- and Self-Modifying Programs 5733
String Handling in Algol 5738
Least-Squares Solution of Systems of Non-Linear Equations by Inverse Methods 5759
Solution of the Two-Dimensional Hidden-Line Problem 5767

Optimization of Extraction Files by Zero-One Programming 5769
Analog Computers
—see Hybrid Computers
Arithmetic
—see Multiplication
Arrays
Storage Array Using Electron Beam Activated Switches 5728
Artificial Intelligence
Memory Circuits Using a Simulated Neuron 5754
Programs Capable of Solving Intellec- tually Difficult Problems 5755
—see also Heuristic Programming, Patterns, Problem Solving, Theorem Proving
Associative
Simulation of a Multiprocessor Using Associative Memories for Executive Control 5721
Automata; TURING MACHINES
Structure and Transition-Preserving Functions of Finite Automata 5712
—see also Sequential

C

Characters, Symbols
On-Line Classification of Handprinted Chinese Characters as a Translation Aid 5751
—see also Patterns

Circuits
—see Logic, Networks, Sequential

Classification
TRACE Time-Shared Routines for Analysis, Classification and Evaluation of Statistical Data 5743
On-Line Classification of Handprinted Chinese Characters as a Translation Aid 5751
—see also Patterns

Codes; Coding; Decoding
Adaptive Coding for Self-Clocking Recording 5766
Order Code for Computer Control 5768
Communications
—see Man-Machine Communications

Compilers; Compiling
Application of Syntax Analysis to Compilation 5737
Use of Transition Matrices in Compiling 5741

Computation
Completeness Results in the Mathematical Theory of Computation 5756

Computational Linguistics
Mathematical Linguistics and Automatic Translation 5748
—see also Languages

Computer Applications
Programs Capable of Solving Intellectually Difficult Problems 5755
Solution of the Two-Dimensional Hidden-Line Problem 5767
Order Code for Computer Control 5768
Definition of Extraction Files and Their Optimization by Zero-One Programming 5769
Use of Hybrid Computing in the Analysis of Steel Rolling 5770
—see also Computational Linguistics, Control, Education, Mathematics, Patterns, Retrieval

Computer Systems
Performance Evaluation of Computing Systems with Memory Hierarchies 5716
Design of Multiple-Access Computer Systems 5720
PL/300 Programming Language for the IBM System/360 Computers 5736
CDL1 Language for Describing Computer Systems 5739
—see also Digital Computers, Operating Systems, Time-Sharing

Computers
Present and Anticipated Developments in the Computer Field 5705
Computer Science in Electrical Engineering Education 5706
Logical Structure of the RC 4000 Multiprogrammed Computer 5719
Microprogrammed Graphical-Interface Computer 5731
Low-Level Data Structure Package for List Processing on the PDP7 Computer 5732
—see also Computer Systems, Digital Computers, Time-Sharing

Consoles
—see Terminals

Control
Simulation of a Multiprocessor Using Associative Memories for Executive Control 5721
Order Code for Computer Control 5768

Converters
Serial Converters Using Charge Rectifiers 5722

Costs
Analytical Cost Comparison of Computer Operating Systems 5746

Cryotron
Switching Speed of Crossed-Film Cryotrons and Their Logic Circuits 5723

D

Data
Low-Level Data Structure Package for List Processing on the PDP7 Computer 5732
TRACE Time-Shared Routines for Analysis, Classification and Evaluation of Statistical Data 5743

Data Processing
—see Computer Systems, Digital Computers, Education, Multiprocessing, Processing

Design
Design of Multiple-Access Computer Systems 5720
—see also Logic Design

Diagnosis
Connection Assignment for Diagnosable Systems 5717
—see also Error, Failures

Differential
Application of Syntax Analysis to Algebraic Differentiation 5737

Digital Computers
Signal Crosstalk in Digital Systems 5718
—see also Computer Systems, Processing

Diodes; Rectifiers
Applications of Charge Rectifiers 5722

Displays
Computer for Linking Multiple Graphical Display Terminals to a Central Processor 5731
Interactive Displays for Document Retrieval 5749
—see also Characters

E

Education
Computer Science in Electrical Engineering Education 5706

Electron Beams; Cathode Rays
Magnetic Film Memory Accessed by Combined Electron and Photon Beams 5727
Electron Beam Activated Switch and Memory 5728

Error
Error Control in the Simplex Technique 5765
—see also Codes, Failures

Executive Systems
Simulation of a Multiprocessor Using Associative Memories for Executive Control 5721
—see also Operating Systems

F

Failures; Faults
Automatic Fault Diagnosis for Systems with Multiple Faults 5717
—see also Diagnosis

Files
Definition of Extraction Files and Their Optimization by Zero-One Programming 5769

Films
—see Magnetic Films

Flow Charts
Output Flow Charts of Self-Modifying Programs 5733

G

Games
Programs for Playing Kalah End Games 5755

Graphics
Microprogrammed Graphical-Interface Computer 5731
Solution of the Two-Dimensional Hidden-Line Problem 5767
—see also Displays

H

Heuristic Programming
Multipurpose Theorem-Proving Heuristic Program 5755

Hybrid Computers
Use of Hybrid Computing in the Analysis of Steel Rolling 5770

I

Information
—see Codes, Retrieval

Integral Equations
Solution of Fredholm Integral Equations by Statistical Estimation 5761

Integration; Integrals
Romberg's Method for Certain Integrals Involving a Singularity 5760

Interconnection
—see Man-Machine Communications

L

Languages; Linguistics
Implementation of Syntax Analysis Using ALGOL 5737
CDL1 Language for Describing Computer Systems 5739
Mechanism for Communicating with a Computer in Natural English 5748
Representation of Syntactic Structure and Meaning of Sentences 5748
—see also Programming Languages

Lasers
High-Density Magneto-Optic Memory with Laser Writing 5726

Lists
Low-Level Data Structure Package for List Processing on the PDP7 Computer 5732

Logic
Effective Utilization of Packaged Logic 5708
Switching Speed of Crossed-Film Cryotrons and Their Logic Circuits 5725
—see also Switching Functions, Threshold

Logic Design
Logical Structure of the RC 4000 Multiprogrammed Computer 5719
Magnetic Films
Cross-Sectional Model for a Thin Magnetic Film Store 5724
Multilayer Processing for Magnetic Film Storage Devices 5725
Magnetic Film Memory Accessed by Combined Electron and Photon Beams 5727

Magnetic Recording
Spectrum Analysis of Digital Magnetic Recording Waveforms 5729
Magneo-Optic
High-Density Magneto-Optic Memory with Laser Writing 5726
Magnetic Film Memory Accessed by Combined Electron and Photon Beams 5727

Man-Machine Communications (Interaction, Systems)
Mechanism for Communicating with a Computer in Natural English 5748
Interactive Displays for Document Retrieval 5749
Man-Machine Communication for On-Line Classification of Chinese Characters 5751

Mathematical Programming
Dynamic Programming Approach to Sequential Pattern Recognition 5750
Solution of Integer Linear Programming Problems by Direct Search 5764
Error Control in the Simplex Technique 5765

Mathematics; Mathematical
Completeness Results in the Mathematical Theory of Computation 5756
—see also Algebra, Differentiation, Integration, Numerical Analysis, Polynomials

Matrices
Use of Transition Matrices in Compiling 5741
Memory
—see Storage
Microprogramming
Microprogrammed Graphical-Interface Computer 5731
Minimization
—see Switching Functions

Models
Cross-Sectional Model for a Thin Magnetic Film Store 5724

Monitor Systems
Evaluation of Executive, Operating and Monitor Systems 5744

Multiple-Access
Design of Multiple-Access Computer Systems 5720
Computer for Linking Multiple Graphical Display Terminals to a Central Processor 5731
—see also Multiprocessing, Multiprogramming, Time-Sharing

Multiplication; Multipliers
Calculation of Mean Shift for Binary Multipliers Using 2, 3, or 4 Bits at a Time 5715

Multiprogramming
Simulation of a Multiprocessor Using Associative Memories for Executive Control 5721
—see also Time-Sharing

Multiprogramming
Logical Structure of the RC 4000 Multiprogrammed Computer 5719

Program Overlay Techniques for Multi-programmed Systems 5747
—see also Time-Sharing

N
Networks
Construction of a Permutation Switching Network 5714
Threshold Logic Network for Recognizing Shape Invariance 5752
—see also Sequential
Neuromime Systems
Memory Circuits Using a Simulated Neuron 5754
—see also Threshold
Noise
Noise in Digital Systems 5718

Numbers
Pseudorandom Number Generation 5763

Numerical Analysis
Numerical Inversion of Laplace Transforms as Fourier Cosine Transforms 5762
—see also Differentiation, Integration, Matrices

On-Line
Bibliographic On-Line Display 5749
On-Line Classification of Handprinted Chinese Characters as a Translation Aid 5751

Operating Systems
Evaluation of Executive, Operating and Monitor Systems 5744
Environment of Computer Operating System Scheduling 5745
Analytical Cost Comparison of Computer Operating Systems 5746
—see also Executive Systems

Optical
—see Lasers, Magneto-Optic

Optimal; Optimization
Optimization of Reduced Dependencies for Synchronous Sequential Machines 5713
Optimization of Extraction Files by Zero-One Programming 5769

P
Patterns
Dynamic Programming Approach to Sequential Pattern Recognition 5750
Threshold Logic Network for Recognizing Shape Invariance 5752
Non-supervised Adaptive Algorithm for Linear Decision Boundaries in Pattern Recognition 5753
—see also Characters
Polynomials
Generalization of Bairwest's Algorithm for Finding Roots of Polynomials 5757

Problem Solving
Programs Capable of Solving Intellectually Difficult Problems 5755

Processing; Processors
Computer for Linking Multiple Graphical Display Terminals to a Central Processor 5731
Time-Sharing versus Batch Processing 5742
—see also Associative, Computers, Multiprogramming

Programming
Grading Subroutine for Student Programming Exercises in the PL/I Language 5707
Optimization of Extraction Files by Zero-One Programming 5769
—see also Heuristic Programming, Multiprogramming

Programing Languages
Grading Subroutine for Student Programming Exercises in the PL/I Language 5707
Basic Concepts of Programming Languages 5734
Method for Defining Programming Languages 5735
Description of EULER, a Generalization of ALGOL 5735
PL/360 Programming Language for the IBM System/360 Computers 5736
Implementation of Syntax Analysis Using ALGOL 5737
String Handling in ALGOL 5738
Construction of Efficient Parsing Algorithms for Programming Languages 5741

Programs
Grading Student Programming Exercises in the PL/I Language 5707
Comparison of Properties of Non- and Self-Modifying Programs 5733
TRACK Time-Shared Routines for Analysis, Classification and Evaluation of Statistical Data 5743

Program Overlay Techniques for Multi-programmed Systems 5747
Programs Capable of Solving Intellectually Difficult Problems 5755

Programs for:
Assignment Problem of Synchronous Sequential Machines 5713
On-Line Classification of Handprinted Chinese Characters as a Translation Aid 5751
Pseudorandom Number Generation 5763

Random Processes
—see Numbers

Real-Time
—see On-Line

Recognition
—see Characters, Patterns

Recording
—see Magnetic Recording

Redundancy
Identification of Symmetry, Redundancy and Equivalence of Boolean Functions 5708

Remote
—see Terminals, Time-Sharing

Retrieval
Interactive Displays for Document Retrieval 5749

S
Scanners
Scan Generators Using Charge Rectifiers 5722

Scheduling
Environment of Computer Operating System Scheduling 5745

Sequential
Optimization of Reduced Dependencies for Synchronous Sequential Machines 5713
—see also Automata
Shift Registers
Shift Registers Using Charge Rectifiers 5722

Simulation
Simulation of a Multiprocessor Using Associative Memories for Executive Control 5721

Software
Evaluation of Executive, Operating and Monitor Systems 5744
—see also Programming, Programs

Statistical
—see also

Solution of Fredholm Integral Equations by Statistical Estimation 5761

Storage; Memory
—see also

Performance Evaluation of Computing Systems with Memory Hierarchies 5716
Serial Memories Using Charge Rectifiers 5722
Cross-Sectional Model for a Thin Magnetic Film Store 5724
Multilayer Processing for Magnetic Film Storage Devices 5725
High-Density Magnetooptic Memory with Laser Writing 5726
Magnetic Film Memory Accessed by Combined Electron and Photon Beams 5727
Electron Beam Activated Switch and Memory 5728
Memory Circuits Using a Simulated Neuron 5754
—see also Associative, Cryotron

Strings
String Handling in ALGOL 5738

Superconductivity
—see Cryotron

Switching; Switches
Construction of a Permutation Switching Network 5714
Switching Speed of Crossed-Film Cryotrons and Their Logic Circuits 5723
Electron Beam Activated Switch and Memory 5728

Switching Functions
Identification of Symmetry, Redundancy and Equivalence of Boolean Functions 5708
Identification of the Type of a Four-Variable Boolean Function 5709
—see also Automata, Logic, Threshold Systems
Signal Crosstalk in Digital Systems 5718
—see also Computer Systems, Operating Systems

T
Terminals
Low-Cost Output Terminal for Time-Shared Computers 5730

Computer for Linking Multiple Graphical Display Terminals to a Central Processor 5731

Theorem Proving
Multipurpose Theorem-Proving Heuristic Program 5755

Threshold
Testing and Realization of Threshold Functions with Don’t Care 5710
Realization Procedure for Multithreshold Threshold Elements 5711
Threshold Logic Network for Recognizing Shape Invariance 5752

Time-Sharing
Low-Cost Output Terminal for Time-Shared Computers 5730
Time-Sharing Versus Batch Processing 5742
TRADE Time-Shared Routines for Analysis, Classification and Evaluation of Statistical Data 5743
Advantages and Disadvantages of Time-Sharing 5746
—see also Multiprogramming

Translation
Mathematical Linguistics and Automatic Translation 5748
On-Line Classification of Handprinted Chinese Characters as a Translation Aid 5751

Translators
Translator Writing Systems 5740

AUTHOR INDEX

A
Abate, J. 5762
Anacker, W. 5716
Ankerlin, R. A. 5709

B
Bavel, Z. 5712
Bertelsen, B. I. 5725
Birtwistle, G. M. 5757
Borko, H. 5749
Brouil, J. K. 5752
Budd, A. E. 5744
Burnbaugh, H. P. 5749
Bursky, P. 5755

C
Cardillo, G. P. 5750
Catt, I. 5718
Chien, R. T. 5717
Chien, Y. T. 5750
Cooper, L. 5764

D
Diephuis, R. J. 5711
Dietmeyer, D. L. 5708
Dubner, H. 5762
Dworsky, L. N. 5723
Dwyer, T. F. 5722

E
Echols, R. E. 5764
Erikson, W. J. 5746
Esa, R. P. 5743
Evans, D. J. 5757
Everhart, T. E. 5728

F
Feldman, J. 5740
Fletcher, R. 5759
Foxley, E. 5737
Freeman, H. 5715, 5767
Fu, K. S. 5750

G
Gabor, A. 5666
Goldberg, N. 5726
Goten, J. W. 5770
Gonzales, R. 5721
Gries, D. 5740, 5741
Groner, G. F. 5751
Gunderson, D. C. 5721

H
Hansen, P. B. 5719
Haring, D. R. 5711
Heafner, J. F. 5751
Hildebrand, J. V. 5763

I
Ide, E. R. 5753

J
Jackson, C. L. 5709
Jutzi, W. 5724

K
Kaplan, D. M. 5756
Kennedy, D. W. 5730
King, P. 5737
Knoll, A. L. 5729
Kuno, S. 5748

L
Loutrel, P. P. 5767

M
MacDonald, N. C. 5728
Metze, G. 5717
Miler, R. 5738

N
Needham, R. M. 5720

P
Pankhurst, R. J. 5747
Preparata, F. P. 5717

R
Ratz, H. C. 5763
Rees, D. 5770
Robinson, T. W. 5751
Rose, G. A. 5731
Rosenberg, R. C. 5730

S
Sackman, H. 5742
Sass, A. R. 5723
Schein, L. B. 5723
Schneider, P. R. 5708
Seppala, Y. 5769
Shahbender, R. 5722
Sheng, C. L. 5710
Shimura, M. 5754
Slagle, J. R. 5755
Smith, B. W. 5707
Smith, D. O. 5727
Smith, D. R. 5752

S
Smith, E. J. 5713
Srinivasan, C. V. 5739
Stefferud, E. A. 5745
Stewart, W. C. 5723
Storoy, S. 5765
Strand, O. N. 5761

T
Temperly, J. F. 5707
Tewarson, R. P. 5758
Timmons, J. A. 5721
Trakhteneerts, E. A. 5768
Tunis, C. J. 5753

W
Waksman, A. 5714
Wang, C. P. 5716
Ware, W. H. 5705
Weber, H. 5735
Weiner, P. 5713
Westwater, E. R. 5761
Wilde, D. U. 5733
Wilkes, M. V. 5720
Wirth, N. 5734, 5735, 5736
Wiseman, N. E. 5732

Y
Yurchenko, V. E. 5768