Graduate weighted least detection occur Dept. pp., can function. 

A non-homogeneous Poisson process is used to model the detection of errors which occur during functional testing of command and control software. The parameters of the detection process are estimated by using a combination of maximum likelihood and weighted least squares methods. Once parameter estimates are obtained, forecasts can be made of cumulative number of detected errors. Forecasting equations of cumulative corrected errors, errors detected but not corrected, and the time required to detect or correct a specified number of errors, are derived from the detected error function. The various forecasts provide decision aids for managing software testing activities. Naval Tactical Data System software error data are used to evaluate several variations of the forecasting methodology and to test the accuracy of the forecasting equations. Due to changes which take place in the actual detected error process, it was found that recent error observations are more representative of future error occurrences than are early observations. Based on a limited test of the model, acceptable accuracy was obtained when using the preferred forecasting method.


A non-homogeneous Poisson process is used to model the detection of errors which occur during functional testing of command and control software. The parameters of the detection process are estimated by using a combination of maximum likelihood and weighted least squares methods. Once parameter estimates are obtained, forecasts can be made of cumulative number of detected errors. Forecasting equations of cumulative corrected errors, errors detected but not corrected, and the time required to detect or correct a specified number of errors, are derived from the detected error function. The various forecasts provide decision aids for managing software testing activities. Naval Tactical Data System software error data are used to evaluate several variations of the forecasting methodology and to test the accuracy of the forecasting equations. Due to changes which take place in the actual detected error process, it was found that recent error observations are more representative of future error occurrences than are early observations. Based on a limited test of the model, acceptable accuracy was obtained when using the preferred forecasting method.

R75-51—Vink, H. A., J. Al, and B. van den Dolder, "Comments on 'The Minimization of TANT Network'" (19 pp., p/a Hr. W. van den Boom; dr. Sillevsilaan 5; Almelo; the Netherlands).

Gimpel has presented a minimization algorithm which realizes a minimal TANT network for any Boolean function, under a Nand gate cost criterion. After the generation of a sufficient set of implicans, a least-cost subset is found from a CC-table (Covering-Closure Table). This paper presents a new reduction rule for the CC-table. This rule introduces closure columns which are multiply implied, instead of singly, as in Gimpel. Moreover, the effect of multiple implication is discussed.


A method for designing the read-only memories needed for multiplication using logarithms is developed. By defining the word length of the multiplicand, multiplier, and product as n bits and the word length of the rounded logarithms as m bits, design curves are given that allow various values of n and m to be selected for a given multiplier accuracy. Then a table is used to determine which combination results in an implementation with the least number of bits.


The Karhunen Loeve [KL] Transform is known to have certain properties which make it optimal for many mean square signal processing applications. Recently, it has been shown that a class of digital images may be presented by a set of boundary value stochastic difference equations in two dimensions. If the boundary conditions of this class of images are fixed, then these equations lead to a fast KL Transform algorithm. Here this fast KL Transform is used for Wiener filtering of images degraded by white or colored noise. Comparisons with Generalized Wiener filtering are made. Recursive filtering equations for implementing the Wiener filter equation are also given and comparisons with other recursive techniques are made. Examples on two dimensional images including both white and colored noise case are given.


In this paper a hypothetical computer is described, and procedures are indicated for showing the correctness of its microprogram. The underlying method used is
that of Birman. However, the computer discussed has some realistic characteristics not shared by the machine treated by Birman, and the details of the microcode validation are complicated by this fact. A formal technique for partitioning the proof is presented and illustrated with examples.

R75-55—Al-Asou, M., A., “Application of Constrained Generalized Inverse to Pattern Classification” (19 pp., Georgia Institute of Technology, Atlanta, Georgia).

In this paper a constrained generalized inverse approach to pattern classification is suggested and motivated statistically. An adaptive constraint procedure is presented and applied to Sebestyen and Edie’s data with very favorable results.


A method is presented that makes it possible to put pictures together into photomosaics that have no visible seams between the pictures.


A general mathematical model is developed as an idealization of the problem of determining cloud motions from satellite pictures. The model consists of superimposed planes of rigid moving polygons. The problem is to determine from a sequence of scenes the linear and angular velocities of the figures, and to decompose the scene into its component figures. Study of the model reveals a number of fundamental relations that form the basis for an analysis program. In particular, a systematic analysis is given of the topological changes that can occur when overlapping figures move together or apart. A computer program based on these results is described, and experimental results are presented.


Algorithms are presented which "compile" an arithmetic expression so that the time required to compute the expression on a computer with many function units (i.e. simple processors) is minimized or made as small as possible with a relatively simple heuristic. The first algorithm presented uses the associative and commutative laws of addition and multiplication to reorder the input expression. The second algorithm also uses distributivity to further reduce computation time. Finally, the effectiveness of the distribution algorithm on polynomials in the inherently serial Horner’s Rule form is shown.

R75-59—Yang, C-C, “Structural Preserving Morphisms of Finite Automata and an Application to Graph Isomorphism” (33 pp., University of Alabama, Birmingham, Alabama).

The transition preserving morphisms (endomorphism, homomorphism, isomorphism and automorphism) of state machines are developed on the basis of nontrivial closed partitions over their state sets. Algorithms with illustrated examples are provided for testing these morphisms. By means of these morphisms, the structural preserving morphisms of finite automata can be readily solved by simply making a constraint on each partition being not only nontrivial and closed but also output-consistent. This paper also suggests two simple methods for solving a very important information retrieval problem, i.e., the graph isomorphism. By both methods, a directed or nondirected graph is transformed to a deterministic Moore type finite automaton with a unity input alphabet. One method is applying the algorithm for determining the automata isomorphisms to define all possible isomorphisms between two given graphs. The other method provides a yes-no answer based on the concept of state compatibility and incomparability. It will be seen that automata theory is fruitful in this regard.

R75-60—Lester, J. M., “Edge Detection in Complex Scenes Based on Gestalt Principles of Proximity and Similarity” (41 pp., University of Wisconsin, Madison, Wisconsin).

An edge detection function is defined on a very broad class of pictures. It departs from previous approaches to edge detection, which are based on differences in intensity or other properties between adjacent areas of a picture, in using the spatial and similarity relations among individual entities, called elements, to determine the locations and probabilities of edges. With auxiliary definition, it may be applied to digitized pictures, treating picture points as elements, or to scenes of diversified shapes distributed in any way—although only scenes of disks are discussed in this paper. This generality permits recursive use of the edge detector in the discrimination of visual textures. An algorithm for the efficient computation of the function is described. The program runs at the University of Wisconsin in a larger system, which includes T.V. and additional programs that cluster edges into boundaries.


In this paper a new set covering problem is considered. The primary concern of the paper is the simplification of the data representation for large-scale Boolean-valued data structures by generating a minimum cover for one of the two data classes. The cover is based on a certain 'neighborhood' concept which can be specified in terms of simple parameters and constructed in the pattern space so that complete pattern separation is accomplished. An optimal set of neighborhoods or a minimum cover is determined which corresponds to the search for an optimal solution for a data representation requiring the minimum number of parameters.

R75-62—Iqbal, B. and K. S. Fu, “A Tree System Approach for Fingerprint Pattern Recognition” (37 pp., Purdue University, West Lafayette, Indiana).

This paper demonstrates how a syntactic approach and, in particular, tree systems can be used to represent and classify fingerprint patterns. The fingerprint impressions are subdivided into sampling squares which are pre- and post-processed for feature extraction. A set of regular tree languages is used to describe the fingerprint patterns and a set data base of trees is used to represent the encoded patterns. In order to infer the structural configuration of the encoded fingerprints, a grammatical inference system is developed. This system utilizes a simple procedure to infer the number of subtrees and relies on a random sample of a computerized interactive technique for the inference of complex structures. 92 fingerprint impressions were used to test the proposed approach. A set of 193 tree grammars was inferred to each sampling square of the 4 x 4 sampling matrix which is capable of generating about 2 x 10^34 classes for the fingerprint patterns.


A directed graph model of parallel computation is introduced. Unlike previously introduced directed and undirected graph models, which were conceived as frameworks for the formal study of parallel computations, this model was conceived to be the basis of a viable general purpose parallel machine. A graph in the model representing a program can be considered the configuration that the machine attains for the execution of this program (graph), i.e., the machine attains the natural structure of the program that it is to execute. Low level parallelism (parallelism within an arithmetic expression, for example) as well as high level parallelism (parallelism among two subroutines, for example) can be exploited very efficiently in the described architecture. The regularity of the organization of the machine makes it very suitable to be implemented using large scale integration.


This correspondence points out the incorrectness of an example microprogram and associated performance improvement as presented in a recent paper. An amended microprogram is presented together with the revised performance improvement.


This paper defines a high-level programming language for a generalized associative array processor (AP). In particular, a data management language for APs, known as D-MAP, is developed for use with a representative AP model. This development model, containing both an associative processing section and a sequential processing section, which may operate concurrently, follows current AP design technology. D-MAP fully exploits the capabilities available in current AP technology by extending ANSI COBOL to support data
The use of a high-level language can significantly facilitate the AP programming process in many applications. However, a search of the literature suggests that few attempts at high-level language implementation, or even definition, have been made to date. The several additional and unique difficulties associated with application-oriented programming of parallel processors in general, and APs in particular, mandates a high-level language capability to produce cost-effective utilization of their processing resources. This paper considers one approach, a first step, toward this development of a high-level language capability to provide some alleviation of immediate programming difficulties as well as provide a tool for further exploration of AP technology.

R75-66—Jones, L. H., K. Carvin, J. Hauser, P. A. Rainey, F. Pherson, J. Reekston, and P. VanNatta, "An Annotated Bibliography on Microprogramming" (17 pp., E. I. du Pont de Nemours & Co., Wilmington, Delaware). This bibliography contains an annotated list of publications (excluding patents) related to microprogramming. Our objective has been to provide a supplement to existing, generally available bibliographies such as those previously published in the SIGMIRC Newsletter ("An Annotated Bibliography on Microprogramming I, Late 1969-Early 1972") by L. H. Jones et al. and The Mitre Bibliography by J. A. Clapp et al. and the bibliographies in Husson, S. S., "Microprogramming: Principles and Practices," Prentice-Hall, Englewood Cliffs (1970) and The Mitre Bibliography. Since both of these bibliographies cover the field from about 1951 until late 1969, most entries in our bibliography are from late 1969 to early 1972.

R75-67—Jones, L. H., and K. Carvin, "Contributed Articles—An Annotated Bibliography on Microprogramming II" (12 pp., E. I. du Pont de Nemours & Co., Wilmington, Delaware). This bibliography contains an annotated list of publications (excluding patents) related to microprogramming. Our objective was to prepare a supplement to existing, generally available bibliographies such as those previously published in the SIGMIRC Newsletter ("An Annotated Bibliography on Microprogramming I, Late 1969-Early 1972") by L. H. Jones et al. and The Mitre Bibliography by J. A. Clapp et al. and the bibliographies in Husson, S. S., "Microprogramming: Principles and Practices," Prentice-Hall, Englewood Cliffs (1970). We have attempted to identify papers which have appeared in earlier versions by giving references to entries in "An Annotated Bibliography on Microprogramming I."

R75-68—Jones, L. H., and M. B. Zeichner, "An Annotated Bibliography on Microprogramming III" (12 pp., E. I. du Pont de Nemours and Co., Wilmington, Delaware). This bibliography contains an annotated list of publications (excluding patents) which are related to microprogramming and have appeared in the past year. It supplements bibliographies previously published in the SIGMIRC Newsletter ("An Annotated Bibliography on Microprogramming I.")


The success of a memory device depends on both technical and economic factors. This paper attempts to analyze and forecast the economic success of computer memory devices through a simple regression model. The results indicate that MOS will dominate the market by 1975 and bubbles will virtually replace all other memory devices by 1980. Market share forecast could be used as a basis of R&D planning.


A new approach for the minimization of switching functions is developed by forming distance and distance-distribution matrices. A new method incorporating the advantages of the map method and the procedures of the tabular method is illustrated and proved. The concept of the order of cube coverings and the least upper bound of cube coverings greatly reduces the effort of finding the prime implicants and more so of finding the minimum sets of prime implicants.


Three algorithms are given for constructing weighted t-ray trees for a given set of integer weighted leaves in which the weight of a parent node is $w_1$ max of its sons. Two goals are of interest; to minimize (A) the number of tree level and (B) the number of internal nodes. Formulas for the optimum values of (A) and (B) are presented in terms of the fan-in and the number and weights of the leaves. It is shown that the first algorithm simultaneously optimizes both (A) and (B), the second only (B) while nearly optimizing (A), and the third optimizes only (A). However the first algorithm cannot be used in dynamic situations in which the number of leaves is not known at the outset; thus the availability of suboptimal procedures is necessary for such situations. Applications are given to parallel processing problems and to switching circuit theory.


We consider the direct solution of the Poisson and biharmonic equations using a number of arithmetic units in parallel. Assume an $n \times n$ grid of mesh points. We show that a finite difference approximation to the Poisson equation can be solved in $O(\log(n))$ time steps using $O(n^2)$ processors in parallel. This holds for Dirichlet, Neumann and periodic boundary conditions. We also show that a finite difference approximation to the biharmonic equation can be solved in $O(n)$ time steps using $O(n^3)$ processors in parallel.


Three procedures of the "pruning" class of transduction procedures for NOR network design by network transformation are discussed. The procedures are used, respectively, on the concepts of "compatible sets of permissible functions," "o-fixed maximum sets of permissible functions" for connections, and "i-fixed maximum sets of permissible functions with gates (CSFP's), (DFMSFP's), and (IFMSFP's)." Used for the latter two guarantees single-irredundant networks.


This thesis discussed the use of macros and other heuristic techniques to increase the speed of backtrack programs. Detailed explanations of several macro programs are given for illustration. These techniques were successful in decreasing execution time by factors of 5 to 10 times over previous non-macro programs.


An interactive COBOL compiler system for the PLATO computer-based education system is described. The system has two major components: an editor and an interpreter. The editor uses language-independent routines developed for PLATO, and the interpreter acts on a representation of the COBOL program to execute the procedural statements. An explanation of the implemented COBOL subset is included.


This report presents two computer algorithms which can find minimal NOR (NAND) networks under fan-in and fan-out restrictions by using the concept of compatible set of permissible functions (CSFP's). These two algorithms realize the network transduction procedures—"merging of gates" and "substitution of gates," which are implemented into computer programs called "NETTRA-G3-FIFO" and "NETTRA-PG1-FIFO," respectively. Both programs have the capabilities in transforming a large non-optimal NOR network under fan-in and fan-out restrictions into a near-optimal one by reducing the number of gates and connections. The paper also includes the results of the computer experiments on these two programs.