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

# Abstracts

**Concepts for Buffer Storage,** C. J. Conti, *Computer Group News* **2,** No. 8, 9–13 (March 1969). The paper discusses concepts for buffering information from backing stores in the computer main frames as exemplified in the IBM System/360 Model 85. The four basic types of buffer design examined are: sector, direct-mapping, fully associative, and set associative buffers. The set associative approach appears to yield the best performing buffer that is practical with today's technology.

**Control Storage Use in Implementing an Associative Memory for a Time-Shared Processor,** L. Hellerman and G. E. Hoernes, *IEEE Transactions on Computers* **C-17,** No. 12, 1144–1151 (December 1968). The Cambridge System, comprising a standard IBM/360 Model 40 and an associative memory, is described from the viewpoint of its implementation using the read-only control storage of the Model 40. In particular, we discuss the use of the ROS in 1) controlling the flow of data between the CPU and the associative memory registers, and 2) handling translation control, absent page indications, and variable field operand pretesting, when the associative memory is used for page translation in a time-sharing mode of operation. Although the main use of the system is as a computer facility that may be shared simultaneously by up to fifteen users, it may also be used as an ordinary batch processor having a small experimental associative memory among its facilities. The magnitude of the ROS additions and modifications, in terms of numbers of microinstructions, is given.

**On Coordination Reduction and Sentence Analysis,** S. R. Petrick and P. S. Rosenbaum, *Communications of the ACM* **12,** No. 4, 223–233 (April 1969). A class of coordination phenomena in natural languages is considered within the framework of transformational theory. To account for these phenomena it is proposed that certain machinery be added to the syntactic component of a transformational grammar. This machinery includes certain rule schemata, the conditions under which they are to be applied, and conditions determining the sequence of subtrees on which they are to be performed. A solution to the syntactic analysis problem for this class of grammars is outlined. Precise specification of both the generative procedure of this paper and its inverse is given in the form of LISP function definitions.

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

*Cornell University, Ithaca, New York

**The Development of the DOS/TOS PL/I Compiler,** W. Thiele, *Elektronische Rechenanlagen* **11,** No. 1, 29–35 (1969). The DOS/TOS PL/I compiler was built for use on smaller machines of the IBM System/360 (starting with Model 25) with a minimum core capacity of 16384. Besides developing the compiler, parts of the task comprise the development of a set of library routines to support execution of the generated programs as well as the definition of a subset of the PL/I language which is adapted to the smaller machine system. This subset of the language must be in line with both the technical and scientific applications as well as commercial applications.

**Dynamic Space-Sharing in Computer Systems,** L. A. Belady and C. J. Kuehner, *Communications of the ACM* **12,** No. 5, 282–288 (May 1969). A formalization of relationships between space-sharing, program behavior, and processor efficiency in computer systems is presented. Concepts of value and cost of space allocation per task are defined and then value and cost are combined to develop a single parameter termed value per unit cost. The intent is to illustrate a possible analytic approach to the investigation of the problems of space-sharing and to demonstrate the method on sample problems.

**Het voorspellen van de betrouwbaarheid van een computersysteem (Organization of Reliability Prediction in a Computer System),** J. Oostingh, *Elektronica en Telecommunicatie* **5,** No. 20, ET 67–73 (May 16, 1969). This paper provides some insight for predicting the reliability of a computer system. Since the computer system is very complex, it is broken down into subsystems, machines and components, not only for the sake of analysis and discussion but also in actual practice. Various methods involving the use of both manual and automatic techniques incorporating reliability feedback into the development/operational-use cycle are described. This ensures that reliability is implemented during the earliest stages and is re-evaluated throughout the life of the computer system.

**A Selected Bibliography on Computer System Performance Evaluation,** prepared by W. Buchholz, *Computer Group News* **2,** No. 8, 21–22 (March 1969). The bibliography contains selected papers on techniques for evaluating computer system performance. Emphasis in the selection was placed on quantitative evaluations using objective and verifiable measures of the work capability of an entire computer system as seen by the user. Reports not generally accessible in the periodical and monograph literature were excluded from the list. Also excluded were papers dealing primarily with qualitative evaluations, measures based on arbitrary judgment factors, design optimization of parts of a system, evaluation of system cost and other economic measures, evaluation of human factors, and trivial observations.

**Series on Operations Research,** *IBM Journal of Research and Development* **13,** No. 4, 342–407 (July 1969). This series of nine papers represents the scope and strategy of operations research, but concentrates on a subgroup of situations that involve the selection or arrangement of discrete elements in a finite system (this feature is an inherent consequence of the nature of the IBM Corporation as compared with, say, the petroleum industry). Networks and matrices dominate the mathematical structure of the problems discussed; these elements are the common ground for transferring experience from one problem situation to another. *Stochastic Model for Manufacturing Cost Estimating* by C. T. Abraham and R. D. Prasad; *Dynamic Inventory Models and Stochastic Programming* by M. N. El Agizy; *Programmed Automatic Customer Engineer (PACE) Dispatch* by

W. H. Evers and S. S. Thakur; *Model of Competition in a Two-seller Market* by D. Savir; *Maximal Biflow in an Undirected Network* by J. C. Arinal; *Algorithm and Average-value Bounds for Assignment Problems* by W. E. Donath; *Shortcut in the Decomposition Algorithm for Shortest Paths in a Network* by T. C. Hu and W. T. Torres; *Minimaximal Paths in Disjunctive Graphs by Direct Search* by J. F. Raimond; *Heuristic Algorithm for the Traveling-salesman Problem* by T. C. Raymond.

**A Statistical Approach to the Computation of Delays in Logic Circuits,** H. D. Schnurmann and K. Maling, *IEEE Transactions on Computers* **C-18,** No. 4, 320–328 (April 1969). This paper describes a method whereby multiple regression techniques are used to predict the delay between input and output signals through a combinatorial logic chain. A delay equation is developed whose variables are chosen to represent the variations found in an actual system environment. First, a technique is described whereby a statistical delay model for a single circuit is developed for measurements on many circuits. The total delay through logic chains of various lengths is predicted by successive application of this model to the elements of the chain. The accuracy of the prediction was checked experimentally, and agreement was found to be very good.

**A Theory of Programming Languages,** R. J. Orgass and F. B. Fitch, *Studium Generale* **22,** No. 2, 113–136 (February 1969). We adopt the point of view that a language is characterized by a relation which relates each well-formed sentence of the language to an appropriate meaning. That is, we equate each language with the semantics of this language. The syntax of such a language is then essentially specified by the domain of this relation, that is, by the class of wffs of the language. We restrict our attention to the class of languages which can be used to program computing machines and obtain a number of properties of these languages.