LABORATORY FOR COMPUTER SCIENCE

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

THE PUBLICATIONS OF THE MIT LABORATORY FOR COMPUTER SCIENCE 1983

545 TECHNOLOGY SQUARE, CAMBRIDGE, MASSACHUSETTS 02139

CREDITS

EDITED BY:	**ROBERT S. CINQ MARS**
LISP CODE BY:	**GEORGE J. CARRETTE and GLENN S. BURKE**
TYPESET BY:	**THE CSA PRESS BEDFORD, MA.**

TABLE OF CONTENTS

TECHNICAL REPORT ABSTRACTS

TR-1
Bobrow, D.G.
NATURAL LANGUAGE INPUT FOR A COMPUTER PROBLEM SOLVING SYSTEM
Pages: 128 Ph.D. Dissertation/September 1964 $7.15
Keywords: natural language, computer problem solving, Student
Abstract: The *STUDENT* problem solving system, programmed in LISP, accepts as input a comfortable but restricted subset of English which can express a wide variety of algebra story problems. *STUDENT* finds the solution to a large class of these problems. *STUDENT* can utilize a store of global information not specific to any one problem, and may make assumptions about the interpretation of ambiguities in the wording of the problem being solved. If it uses such information, or makes any assumptions, *STUDENT* communicates this fact to the user.

The thesis includes a summary of other English language question-answering systems. All these systems, and *STUDENT*, are evaluated according to four standard criteria.

The linguistic analysis in *STUDENT* is a first approximation to the analytic portion of a semantic theory of discourse outlined in the thesis. *STUDENT* finds the set of kernel sentences which are the base of the input discourse, and transforms this sequence of kernel sentences into a set of simultaneous equations which form the semantic base of the *STUDENT* system. *STUDENT* them tries to solve this set of equations for the values of requested unknowns. If it is successful it gives the answers in English. If not, *STUDENT* asks the user for more information, and indicates the nature of the desired information. The *STUDENT* system is a first step toward natural language communication with computers. Further work on the semantic theory proposed should result in much more sophisticated systems. {AD 604-730}

TR-2
Raphael, B.
SIR: A COMPUTER PROGRAM FOR SEMANTIC INFORMATION RETRIEVAL
Pages: 169 Ph.D. Dissertation/June l964 $8.40
Keywords: semantic information retrieval, natural language
Abstract: *SIR* is a computer system, programmed in the LISP language, which accepts information and answers questions expressed in a restricted form of English. This system demonstrates what can reasonably be called an ability to "understand" semantic information. *SIR's* semantic and deductive ability is based on the construction of an internal model, which uses word associations and property lists, for the relational information normally conveyed in conversational statements.

A format-matching procedure extracts semantic content from English sentences. If an input sentence is declarative, the system adds appropriate information to the model. If an input sentence is a question, the system searches the model until it either finds the answer or determines why it cannot find the answer. In all cases *SIR* reports its conclusions. The system has some capacity to recognize exceptions to general rules, resolve certain semantic ambiguities, and modify its model structure in order to save computer memory space.

Judging from its conversational ability, *SIR* is more "intelligent" than any existing question-answering system. The author describes how this ability was developed and how the basic features of *SIR* compare with those of other systems.

The working system, *SIR*, is a first step toward intelligent machine communication. The author proposes a next step by describing how to construct a more general system which is less complex and yet more powerful than *SIR*. This proposed system contains a generalized version of the *SIR* model, a formal logical system called *SIR1*, and a computer program for testing the truth of *SIR1* statements with respect to the generalized model by using partial proof procedures in the predicate calculus. The thesis also describes the formal properties of *SIR1* and how they relate to the logical structure of *SIR*. {AD 608-499}

TR-3
Corbató, F.J.
SYSTEM REQUIREMENTS FOR MULTIPLE ACCESS, TIME-SHARED COMPUTERS
Pages: 14 May l964 $3.75
Keywords: time-shared computers
Abstract: It is now clear that is is possible to create a general-purpose time-shared multiple access system on most contemporary computers. However, it is equally clear that none of the existent computers are well designed for multiple access systems. At present, good service to a few dozen simultaneous users is considered state-of-the-art.

Discussions include: clocks, memory protection and supervisor mode, program relocation and common subroutines which expose the reader to the difficulties encountered with contemporary machines when multiple user multiple-processor systems are considered. {AD 608-501}

TR-4
Ross, D.T., Feldman C.G.
VERBAL AND GRAPHICAL LANGUAGE FOR THE AED SYSTEM; A PROGRESS REPORT
Pages: 26 May 1964 $4.10
Keywords: computer-aided design, time-sharing
Abstract: For Computer-Aided Design use of time-sharing a single language which can take either verbal or graphical form is required. This paper describes how a single language processing technique, which is in turn a special application of more general concepts concerning the step-by-step growth and processing of large structures of interrelated elements, can efficiently process both language forms in the same manner. Illustrations of the concepts involved are also drawn from the methods used in the AED-O Compiler, an efficient ALGOL-60-based compiler used in Computer-Aided Design work, which is available as a public command in the Project MAC CTSS. {AD 604-678}

TR-6
Biggs, J.M., Logcher, R.D.
STRESS: A PROBLEM-ORIENTED LANGUAGE FOR STRUCTURAL ENGINEERING
Pages: 22 May l964 $4.00
Keywords: STRESS, structure analysis
Abstract: *STRESS* is a general purpose programming system for the analysis of structures. Compared to most other structural programs it has three distinguishing characteristics: (1) The input language is that of the structural engineer which makes possible direct communication between the engineer and the machine; (2) The system is capable of analyzing a wide variety of structural types and loading conditions thus permitting industrial use on a routine basis; and (3) The design process is expedited by the fact that modifications of the original structure for alternate designs can be easily executed. This last capability is most effective when *STRESS* is used in the time-sharing mode. These features combine to provide a system which not only reduces the effort required for structural analysis but, more significantly, enhances the designer's ability to evolve an efficient structure. {AD 604-679}

TR-7
Weizenbaum, J.
OPL-I AN OPEN ENDED PROGRAMMING SYSTEM WITHIN CTSS
Pages: 9 April 1964 $3.60
Keywords: OPL-I, CTSS, time-shared computers
Abstract: OPL-I, an incremental programming system presently operating with CTSS, permits the user to augment both his program and his data base during widely separated successive sessions at his terminal. Facilities are provided which make it possible for the user to operate on his already established data base both by means of built-in operators and in terms of operators (functions) which the user has previously defined in the language of the system. Underlying the system is a powerful list processing scheme embedded in FORTRAN (SLIP). The machinery of this fundamental language drives the system and is also largely available to the user. The data base generated by the user is therefore a set of list structures (trees), and most of the operators available to him are list processing operators. Data structures with considerably complex inter-relational properties may therefore be treated quite directly. {AD 604-680}

TR-8
Greenberger, M.
THE OPS-1 MANUAL
Pages: 170 May l964 $8.45
Keywords: time-sharing, OPS-1, management applications, simulation methodology, data analysis

Abstract: The recent attainment and continuing development of personally accessible computer facilities have opened another chapter in the use of machines by man. A number of current research efforts, including Project MAC at M.I.T., are designing new conceptual systems to adapt the emerging technology to a wide range of human activity. Activities relating to management are the concern of a trial system at Project MAC called OPS-1. The OPS-1 system and the experiment that launched it are described in this manual. {AD 604-681}

TR-11
Dennis, J.B.
PROGRAM STRUCTURE IN A MULTI-ACCESS COMPUTER
Pages: 16 May 1964 $3.80
Keywords: processing units, main memory, multiple-access computers, terminal devices
Abstract: A multi-access computer (MAC) system consists of *processing units* and directly addressable *main memory* in which procedure information is interpreted as sequences of operations on data, a system of *terminal devices* through which users may communicate with procedures operating for them, and mass memory where procedures and data may be held when not required for immediate reference. One fundamental attraction of the MAC concept is the increased productivity of "computer catalyzed research" that results from close man-machine interaction. Another attraction is wealth of data and procedures that are accessible to a large user community through the file memory of a MAC system. In this report thoughts are developed which form an adequate model of program structure. These concepts have grown out of many discussions with colleges in Project MAC, and our experience to date in the design and operation of multi-access computer systems. {AD 604-500}

TR-12
Fano, R.M.
THE MAC-SYSTEM: A PROGRESS REPORT
Pages: 24 October 1964 $4.05
Keywords: MAC, CTSS, time-shared computers, machine-aided cognition
Abstract: The notion of machine-aided cognition implies an intimate collaboration between a human user and a computer in a real-time dialogue on the solution of a problem, in which the two parties contribute their best capabilities. In order for this intimate collaboration to be possible, a computer system is needed that can serve simultaneously a large number of people, and that is easily accessible to them, both physically and intellectually. The present MAC System is a first step toward this goal. The purpose of this paper is to present a brief description of the current system, to report on the experience gained from its operation, and to indicate directions along which future developments are likely to proceed. {AD 609-296}

TR-13
Greenberger, M.
A NEW METHODOLOGY FOR COMPUTER SIMULATION
Pages: 27 October 1964 $4.15
Keywords: simulation
Abstract: Computer simulation is a cooperative venture between researcher and information processor, but the processor's role customarily begins too late. The researcher can benefit substantially by bringing the computer up into the earlier, creative phases of the simulation process. An on-line computer system that makes this possible is described. {AD 609-288}

TR-14
Roos, D.
USE OF CTSS IN A TEACHING ENVIRONMENT
Pages: 29 November 1964 $4.20
Keywords: CTSS, time-sharing, education
Abstract: Computer time-sharing offers many interesting possibilities for use in teaching computer technology. It might be expected that with proper hardware and software, students using time-sharing as a teaching machine could acquire proficiency in the fundamentals of programming more easily than using batch-processing.

To test this hypothesis, the M.I.T. Department of Civil Engineering divided a freshman programming class, so that half the students used batch-processing methods, and half used the Project MAC time-sharing system to do the same work.

This paper describes the experiment and its tentative results. {AD 661-807}

TR-16
Saltzer, J.H.
CTSS TECHNICAL NOTES
Pages: 77 March 1965 $5.65
Keywords: multiple-access computers, time-shared computers, on-line computer systems
Abstract: This report is a technical description of the 7094 Compatible Time-Sharing System in use at Project MAC and the M.I.T. Computation Center. It is designed to acquaint a system programmer with the techniques of construction which were used in this particular time-sharing system. Separate chapters discuss the overall supervisor program flow; console message input and output; the scheduling and storage algorithms; and a thumbnail sketch is given of each of the subroutines which make up the supervisor program.

This report was prepared with the aid of the compatible time-sharing system and the *TYPSET* and *RUNOFF* commands. {AD 612-702}

TR-17
Samuel, A.L.
TIME-SHARING ON A MULTICONSOLE COMPUTER
Pages: 23 March 1965 $4.00
Keywords: time-sharing, CTSS
Abstract: After a brief historical review and a description of the three basic types for time-sharing systems, the general purpose time-sharing system as exemplified by the M.I.T. CTSS system is described in general terms, with particular attention to the way the system looks to the user. {AD 462-158}

TR-18
Scherr, A.L.
AN ANALYSIS OF TIME-SHARED COMPUTER SYSTEMS
Pages: 178 Ph.D. Dissertation/June 1965 $8.65
Keywords: machine-aided cognition, multiple-access computers, on-line computer systems, time-sharing
Abstract: Some of the aspects of the operation of time-shared, interactive computer systems are analyzed. The emphasis is on the reaction of hardware systems to the demands that its users make upon it. Simply shared systems and their users in order to be able to predict the performance of the two operating together. Portions of this problem include the specification and measurement of user characteristics, the development and verification of both simulation and mathematical models for time-shared systems, and the specification and measurement of performance metrics for such systems. The user and some of the performance measurements were made on Project MAC's "Compatible Time-Sharing System" (CTSS).

First, simulation models are used to study the effects of changing small details in the operation of CTS-like systems. Then, a continuous-time Markov process model is derived to predict the performance of a broad class of systems. Throughout, the CTSS data are used as a basis for comparison with model predictions. In order to be able to take measurements and to build models, many definitions of commonly used time-shared system terminology are made precise. {AD 470-715}

TR-19
Russo, F.J.
A HEURISTIC APPROACH TO ALTERNATE ROUTING IN A JOB SHOP
Pages: 44 S.B. & S.M. Thesis/June 1965 $4.65
Keywords: alternate routing, machine-aided cognition, multiple-access computers, on-line computer systems, real-time systems, time sharing
Abstract: The research reported here investigates the use of heuristics for selecting from several alternate routes resulting from partially ordered tasks in a job shop order file. The experimental vehicle employed was digital simulation.

The concept of the "alternate string" has been developed to generalize the existence of partially ordered operations. That term is defined as a concatenation of operations that can be performed in any order, with the additional specification that all within the string must be completed before any operation past the string can be attempted. The presence of alternate strings with two or more member gives rise to the alternate routing problem, whose solution is approached by heuristic methods.

Choosing from among several alternate routes constitutes a three level decision problem. At the lowest level, routes can be chosen when the order enters the shop. This is equivalent to fixed routing. At a higher level, alternates can be selected at the time of transition from one work station to another. The third decision level occurs at operation time, when one of the alternate operations is placed on a machine. Heuristics were tested at the latter two levels.

There were two prior assertions that this thesis set out to prove. The first was that alternate routing at the highest decision level would produce significant reductions in the mean tardiness of orders completed past their designated due dates, the improvement being both relative to fixed routing and to alternate routing heuristics implemented at lower decision levels. Secondly, the contention was made that the improvement would be os such a magnitude that on-line, real-time systems become economically justifiable as a means of mitigation the attendant control problems caused by non-deterministic paths through the queuing network.

The methodology employed here was to conduct two passes of simulated shop runs. The first, with two artificially high levels of alternate incidence, tested the efficiency of five different alternate routing heuristics in reducing mean tardiness. The second pass consisted of runs with the best heuristic developed during the first experimental phase applied to a realistic length and frequency of alternate strings.

The results of the experiments strongly support the assertions made at the outset of the thesis. The performance characteristics of the different heuristics are discussed at length. In addition, some implications are drawn of the computational nature of alternate routing and the difficulties encountered in implementing alternate routing heuristics at operation time. {AD 474-018}

TR-20
Wantman, M.E.
CALCULAID: AN ON-LINE SYSTEM FOR ALGEBRAIC COMPUTATION AND ANALYSIS
Pages: 53 S.M. Thesis/September 1965 $4.90
Keywords: numerical manipulations, numerical calculations
Abstract: OPS is an on-line system developed by M. Greenberger et al. at Project MAC. The present work provides a powerful and simple way to perform numerical manipulations and calculations within OPS. The program package is called CALCULAID.

A method of executing algebraic assignment statements, of which MAD and FORTRAN assignments are a subset, is provided. When this assignment-statement ability is coupled with other features of the OPS system, such as unconditional transfers, general conditionals, and array and function declarations, most of the ability of a compiler language is provided. Because the programs written in OPS are executed interpretively, OPS-3 programs can be changed and re-run immediately, without being compiled.

The other elements of CALCULAID are a program for creating multiple linear regression models, rank-ordering and counting data, and finding roots to polynomial equations in one unknown.

The applications of CALCULAID to the analysis of a round-robin scheduling model and to a process-control problem are discussed, and conclusions regarding the suitability of running computational programs in an interpretive mode are drawn. {AD 474-019}

TR-21
Denning, P.J.
QUEUEING MODELS FOR FILE MEMORY OPERATION
Pages: 106 S.M. Thesis/October 1965 $6.50
Keywords: machine-aided cognition, multiple-access computers, on-line computer systems, queueing models, real-time systems, time-sharing
Abstract: A model for the auxiliary memory function of a segmented, multiprocessor, time-shared computer system is set up. A drum system in particular is discussed, although no loss of generality is implied by limiting the discussion to drums. Particular attention is given to the queue of requests waiting for drum use. It is shown that a shortest access time first queue discipline is the most efficient, with the access time being defined as the time required for the drum to be positioned, and is measured from the finish of service of the last request to the beginning of the data transfer for the present request. A detailed study of the shortest access time queue is made, giving the minimum access time probability distribution, equations for the number in the queue, and equations for the wait in the queue. Simulations were used to verify these equations; the results are discussed. Finally, a general Markov Model for Queues is discussed in an Appendix. {AD 624-943}

TR-22
Greenberger, M.
THE PRIORITY PROBLEM
Pages: 28 November 1965 $4.15
Keywords: priority problem, time-sharing, scheduling
Abstract: Priority decisions arise whenever limited facilities must be apportioned among competitive demands for service. Broadly viewed, even the familiar first-come-first-served discipline is a priority rule. It favors the longest-waiting user, and guards against excessive delays. Other priority rules, such as shortest-job-next, are keyed instead to considerations of operating efficiency. Urgency of request is still another common consideration. Since these considerations often conflict, the priority rule serves as mediator. Use of a common cost measure can help effect this mediation, as results from recent job-shop simulations illustrate.

A priority operation of contemporary interest is scheduling a time-shared computer among its concurrent users. Service requirements are not known in advance of execution. To keep response times short for small requests, service intervals are partitioned and segments are served separately in round-robin fashion. A mathematical analysis pinpoints the tradeoff between overhead and discrimination implicit in this procedure, and allows alternate strategies to be costed. Extensions of the simple round-robin procedure are suggested, the objectives of time-sharing are reviewed, and implications are drawn for the design of future priority and pricing systems. {AD 625-728}

TR-23
Dennis, J.B., Van Horn, E.C.
PROGRAMMING SEMANTICS FOR MULTIPROGRAMMED COMPUTATIONS
Pages: 46 December 1965 $4.70
Keywords: time sharing, parallel programming, programming semantics, multiprogramming
Abstract: The semantics are defined for a number of meta-instructions which perform operations essential to the writing of programs in multiprogrammed computer systems. These meta-instructions relate to parallel processing, protection of separate computations, program debugging, and the sharing among users of memory segments and other computing objects, the names of which are hierarchically structured. The language sophistication contemplated is midway between an assembly language and an advanced algebraic language. {AD 627-537}

TR-24
Kaplow, R., Strong, S., Brackett, J.
MAP: A SYSTEM FOR ON-LINE MATHEMATICAL ANALYSIS
Pages: 103 January 1966 $6.40
Keywords: graphical input-output, man-machine interaction, mathematics, multiple-access computers, numerical analysis, time sharing
Abstract: This manual describes a computer suitable for use on the time-sharing facility at the M.I.T. Computation Center or at Project MAC. Designed for direct computer access through a remote console, the system replaces the normal procedures of programming with a question and answer interchange between the user (hereinafter called U) and the computer (hereinafter called C). The system is intended for the solution of mathematical problems. It should be usable by a person with no knowledge of computers or programming and little knowledge of numerical analysis. Within its range of capabilities, it should be as efficient as are the normal means of computer access for the more sophisticated user.

The system establishes a "conversation" between U and C with an electric typewriter as the means of communication. U can give information to C and can ask it certain questions. C can answer those questions if it is given enough information. C can also ask questions and can therefore request any missing information. In addition, C can explain procedures to U in order to help the latter transmit the required information in a proper form. U, therefore, only needs to know a few basic rules, such as how to phrase his questions and how to name and tabulate his data. {AD 476-443}

TR-25
Stratton, W.D.
INVESTIGATION OF AN ANALOG TECHNIQUE TO DECREASE PEN-TRACKING TIME IN COMPUTER DISPLAYS
Pages: 64 S.M. Thesis/March 1966 $5.25
Keywords: light pens, pen tracking, time-sharing
Abstract: Many modern digital computer systems contain cathode-ray-tube display equipment to facilitate man-machine communications. Through the use of a display and a light-sensitive pen, graphical material can be directly

inserted into the computer by using the pen to control the position of the electron beam at the face of the CRT-- a process called pen tracking. Beam position is continually sampled by the computer, permitting continuous display of the material being sketched. In present digital pen-tracking techniques, a tracking pattern (usually a cross) with a substantial number of points is generated on the face of the CRT and the binary response of the pen to the individual points of the pattern is employed to calculate pen position. The large number of pattern points, and the phosphor decay time associated with each, yield a typical tracking cycle of 500 to 1000 microseconds. Since the cycle must be repeated about 100 times per second, 5 to 10 percent of display time is consumed.

To reduce the time required by the tracking operation, an analog technique employing a four-point tracking pattern is proposed in this study, in which the amplitude response of the pen to corresponding pairs of points is used to determine the position of the pen relative to the center of the pattern. To study the method, one channel of the proposed two-channel analog tracking system was designed, constructed, and coupled to the horizontal channel of a high-speed computer display console. To avoid the phosphor-decay limitation, an experimental "beam" pen capable of detecting the electron beam rather than the phosphor luminescence is employed. The system includes a pattern generator, sample-and-hold gates, difference amplifier, envelope detector and noise filter, and a threshold-logic analog-to-digital converter. The time required to generate the tracking pattern and develop the binary equivalent of the horizontal distance separating pen and pattern center is only 25 microseconds. Tracking is generally satisfactory, but some anomalies were noted, apparently due to the characteristics of the experimental pen being used.

It is concluded that the analog technique is feasible for improving the speed of pen tracking, but recommended that further studies be made of the limitations inherent in the method. {AD 631-396}

TR-26
Cheek, T.B.
DESIGN OF A LOW-COST CHARACTER GENERATOR FOR REMOTE COMPUTER DISPLAYS
Pages: 61 S.M. Thesis/March 1966 $5.15
Keywords: character generation, computer displays, time-sharing
Abstract: A requirement exists for a low-cost remote display terminal with alphanumeric and line-drawing capabilities for use with time-shared computer systems. This thesis, conducted as part of the overall remote display design project, was undertaken to investigate novel approaches to character generation, with the goal of drastically reducing present-day costs for such devices.

A survey of existing devices and character generation techniques was carried out, and a design approach was chosen which takes advantage of mass-fabrication techniques. This includes using a five-by-seven dot matrix raster and a resistor array "read-only" character memory for the 96 printable symbols of the Revised Proposed ASCII Code. Circuits designed, included a dot matrix generator and a register array memory with selection logic sense amplifiers, and a shift register output buffer. An experimental character generator with an eight-word memory was built, largely using integrated circuits and was found to work as desired. It is concluded that the design approach will yield a character generator that is of low enough cost to find wide use in remote computer terminals. {AD 631-269}

TR-27
Edwards, D.j.
OCAS — ON-LINE CRYPTANALYTIC AID SYSTEM
Pages: 54 S.M. Thesis/March 1966 $4.95
Keywords: cryptography, cryptology, security
Abstract: Deficiencies of various programming languages for dealing with quantities frequently encountered in cryptanalysis of simple cipher systems will be discussed. A programming system is proposed which will permit a cryptanalyst to write and debug programs to aid in the solution of cryptograms or crytographic systems. The basic elements of the proposed programming system are discussed in detail. They include: 1) a programming language to handle both algebraic quantities and character strings, 2) a display generator to permit quick specification of a display frame containing both alphanumeric strings and numerical data for an on-line CRT display device, and 3) an on-line program to control operation of the system and in debugging programs written in the proposed language. {AD 633-678}

TR-28
Smith, A.A.
INPUT/OUTPUT IN TIME-SHARED, SEGMENTED, MULTIPROCESSOR SYSTEMS
Pages: 73 S.M. Thesis/June 1966 $5.50
Keywords: input/output, segmented, multiprocessor systems, time-shared
Abstract: After introducing and defining the concepts of time-sharing, segmentation, and multiprocessing, two classes of systems incorporating these are introduced. Both classes use associative memories, as 'look behind' devices to speed the operation of addressing the segmented memory, with the distinction between classes being the location of the associative memory. In one class, there is one associative memory for each processing element, no matter how many main memory units are connected to a processor; in the second class, there is one associative memory for each main memory unit, with the processors sharing associative memory. After introducing two criteria for input/output systems, that the overhead associated with their use be small and that they may be physically and logically simple, and describing further operations of the systems, it is concluded that members of the second class, having shared associative memories, best meet there criteria. {AD 637-215}

TR-29
Ivie, E.L.
SEARCH PROCEDURES BASED ON MEASURES OF RELATEDNESS BETWEEN DOCUMENTS
Pages: 240 Ph.D. Dissertation/June 1966 $10.55
Keywords: document searching, information retrieval, clusters, file structure, relatedness
Abstract: In this thesis a new type of information retrieval system is suggested which utilizes data of the type generated by the users of the system instead of data generated by indexers.

The theoretical model on which the system is based consists of three basic elements. The first element is a measure of the relatedness between document-pairs. It is derived from information theory. The second element is a definition of what constitutes a set (cluster) of inter-related documents. This definition is based on the measure of relatedness. The last element is a procedure which transforms a request for information into a cluster of answer documents.

Requests are made by designating one or more documents to be of interest and perhaps some to be of no interest. The requester can continue to interact with the procedure as it locates the answer cluster by specifying as interesting or not interesting other documents which are presented to him. The answer cluster which is generated is automatically made as small (specific) or as large (general) as is desired, depending on the initial request and the subsequent interactions.

An experimental system was developed to test the model in a realistic environment. It was programmed for the Project MAC time-sharing system and utilized the physics data file of the Technical Information Project. Citations were used as the data base for the measure of relatedness. A file structure and retrieval language were designed which allowed close man-machine coupling.

Experiments were conducted which compared the clusters of documents produced by the experimental system with various sets of documents of known mutual pertinence. These sets included bibliographies from review articles, subject categories, and sets of documents found to be of interest to selected users of the system. It was found that between 60-90% of the documents of known pertinence were included in the corresponding clusters. Ways of improving this retrieval efficiency even further are suggested. {AD 636-275}

TR-30
Saltzer, J.H.
TRAFFIC CONTROL IN A MULTIPLEXED COMPUTER
Pages: 79 Sc.D. Thesis/July I966 $5.70
Keywords: computation utility systems, utilities, time-sharing, Multics
Abstract: This thesis describes a scheme for processor multiplexing in a multiple user, multiple processor computer system. The scheme is based upon a distributed supervisor which may be different for different users. The processor multiplexing method provides smooth inter-process communication, treatment of input/output control as a special case of inter-process communication, and provision for a user to specify parallel processing or simultaneous input-output without interrupt logic. By treatment of processors in an anonymous pool, smooth and automatic scaling of system capacity is obtained as more processors and more users are added. The basic design has intrinsic overhead in processor time and memory space which remains proportional to the amount of useful work the system

does under extremes of system scaling and loading. The design is not limited to a specific hardware implementation; it is intended to have wide application to multiplexed, multiple processor computer systems. The processor traffic controller described here is an integral part of **Multics**, a **Multi**plexed **I**nformation and **C**omputing **S**ervice under development by Project MAC at M.I.T., in cooperation with the Bell Telephone Laboratories and the General Electric Company. {AD 635-966}

TR-31
Smith, D.L.
MODELS AND DATA STRUCTURES FOR DIGITAL LOGIC SIMULATION
Pages: 145 S.M. Thesis/August 1966 $7.70
Keywords: digital logic simulation, time-sharing, machine-aided cognition, circuit and signal models
Abstract: A digital logic simulation system is proposed for design verification. Logic to be simulated is specified with a high level register transfer design language, and the simulation system operates on-line on a large time-shared computer. The problem of selecting adequate circuit and signal models for this purpose is considered. Models are proposed with sufficient timing detail to allow the simulation system to detect timing errors which currently are found by manual checking or prototype debugging.

A data structure for representing idealized circuit and signal models and a matching simulation algorithm is discussed. The data structure is a direct representation of a complete subset of the design language and is organized so that it can be incrementally modified to reflect design changes. The simulation algorithm is very efficient because combinational levels are re-evaluated only if their values are needed and may have changed since last evaluated.

The data structure is expanded to represent detailed circuit and signal models. A method of intermixing idealized and detailed models and efficiently simulating very large designs is discussed. Extensions are proposed to the design language so that it can be used to specify model parameters and serve as the simulation command language. {AD 637-192}

TR-32
Teitelman, W.
PILOT: A STEP TOWARDS MAN-COMPUTER SYMBIOSIS
Pages: 193 Ph.D. Dissertation/September 1966 $9.10
Keywords: artificial intelligence, computer problem solving, heuristic programming, man-machine interaction, time-sharing
Abstract: *PILOT* is a programming system constructed in LISP. It is designed to facilitate the development of programs by easing the familiar sequence: write some code, run the program, make some changes, write some more code, run the program again, make some changes, write some more code, rune the program againe, etc. As a program becomes more complex, making these changes becomes harder and harder because the implications of changes are harder to anticipate.

In the *PILOT* system, the computer plays an active role in this evolutionary process by providing the means whereby changes can be effected immediatly, and in ways that seem natural to the user. The user of *PILOT* feels that he is giving advice, or making suggestions, to the computer about the operation of his programs, and that the system then performs the work necessary. The *PILOT* system is thus an interface between the user and his program, monitoring both the requests of the user and the operation of his program.

The user may easily modify the *PILOT* system itself by giving it advice about its own operation. This allows him to develop his own language and to shift gradually onto *PILOT* the burden of performing routine but increasingly complicated tasks. In this way, he can concentrate on the menial tasks of editing, rewriting, or adding to his programs. Two detailed examples are presented.

PILOT is a first step toward computer systems that will help man to formulate problems in the same way they now help him to solve them. Experience with it supports the claim that such "symbiotic systems" allow the programmer to attack and solve more difficult problems. {AD 638-446}

TR-33
Norton, L.M.
ADEPT - A HEURISTIC PROGRAM FOR PROVING THEOREMS OF GROUP THEORY
Pages: 178 Ph.D. Dissertation/October 1966 $8.65
Keywords: group theory, theorem-proving
Abstract: A computer program, named *ADEPT* (*A D*istinctly *E*mpirical *P*rover of *T*heorems), has been written which proves theorems taken from the abstract theory of groups. Its organization is basically heuristic, incorporating many of the techniques of the human mathematician in a "natural" way. This program has proved almost 100 theorems, as well as serving as a vehicle for testing and evaluating special-purpose heuristics. A detailed description of the program is supplemented by accounts of its performance on a number of theorems, thus providing many insights into the particular problems inherent in the design of a procedure capable of proving a variety of theorems from this domain. Suggestions have been formulated for further efforts along these lines, and comparisons with related work previously reported in the literature have been made. {AD 645-660}

TR-34
Van Horn, E.C.
COMPUTER DESIGN FOR ASYNCHRONOUSLY REPRODUCIBLE MULTIPROCESSING
Pages: 237 Ph.D. Dissertation/November 1966 $10.45
Keywords: automata, programming languages, coordinated multiprocessing
Abstract: A concept is presented for designing either a computing system, or a programming language system, so that the following problem is avoided: during a multiprocess computation in which several processes communicate, and in which the relative timing of the performance of the processes is arbitrary, the output produced by the computation might not be a function of only the initial computation state, i.e., of only the inputs and initial program of the computation. The design concept for avoiding this problem is explained by defining an apparently new class of abstract machines called machines for coordinated multiprocessing, or MCM's. Processes are coordinated in an MCM by means of a count matrix, which may be modified by actions of processes, and which determines the processes enabled to proceed at any instant. Remarks are made to suggest that a computing facility which behaves like an MCM can be both constructed and programmed at reasonable cost. It is proved that every MCM has the properties output functionality and output assuredness. Output functionality means that each symbol in every output stream is a function only of the initial computation state. Output assuredness means that for each output stream the maximum number of symbols produced in the stream, or the fact that the number of such symbols has no upper bound, is a function only of the initial computation state. {AD 650-407}

TR-35
Fenichel, R.R.
AN ON-LINE SYSTEM FOR ALGEBRAIC MANIPULATION
Pages: 111 Ph.D. Dissertation/December 1966 $6.65
Keywords: algebraic manipulation, machine-aided cognition, time-sharing
Abstract: This thesis describes an approach to the problem of programming a computer for algebraic manipulation. The motivating threads of the work are first picked up in Chapter I.

To test the descriptive intuitions urged normatively in Chapter I, an experimental system was actually implemented. This system is described in Chapter II and in the Appendices.

The system was variously exercised, as reported in Chapters III, IV, and V. In addition to certain examples, Chapter III includes a more speculative discussion of the range of the system.

The exercises chosen for Chapters IV and V (algebraic "simplification" and "limit problems," respectively) proved to be worthy of some discussion not related to the system under test.

Finally, Chapter VI is a mass of hindsight, reconsideration, and evaluation. On the basis of the experience described in preceding chapters, future directions of work are suggested. {AD 657-282}

TR-36
Martin, W.A.
SYMBOLIC MATHEMATICAL LABORATORY
Pages: 336 Ph.D. Dissertation/January 1967 $13.40
Keywords: graphic display, light pens, non-numerical analysis, symbolic manipulation
Abstract: A large computer program has been developed to aid applied mathematicians in the solution of problems in non-numerical analysis which involve tedious manipulations of mathematical expressions. The mathematician uses typed commands and a light pen to direct the computer in the application of mathematical transformations; the intermediate results are displayed in standard text-book format so that the system user can decide the next step in the problem solution.

Three problems selected from the literature have been solved to illustrate the use of the system. A detailed analysis of the problems of input, transformation, and display of mathematical expressions is also presented. {AD 657-283}

TR-37
Guzman-Arenas, A.
SOME ASPECTS OF PATTERN RECOGNITION BY COMPUTER
Pages: 118 S.M. Thesis/February 1967 $6.85
Keywords: machine-aided cognition, multiple-access computers, object identification, pattern recognition, symbolic manipulation
Abstract: A computer may gather a lot of information from its environment in an optical or graphical manner.

A scene, as seen for instance from a TV camera or a picture, can be transformed into a symbolic description of points and lines or surfaces. This thesis describes several programs, written in the language *CONVERT*, for the analysis of such descriptions in order to recognize, differentiate and identify desired objects or classes of objects in the scene. Examples are given in each case.

Although the recognition might be in terms of projections of 2-dim and 3-dim objects, we do not deal with stereoscopic information.

One of our programs (Polybrick) identifies parallelepipeds in a scene which may contain hidden bodies and non-parallelepipedic objects. The program TD works mainly with 2-dimensional figures, although under certain conditions successfully identifies 3-dim objects. Overlapping objects are identified when they are transparent.

A third program, DT, works with 3-dim and 2-dim objects, and does not identify objects which are not completely seen.

Important restrictions and suppositions are: (a) the input is assumed perfect (noiseless), and in a symbolic format; (b) no perspective deformation is considered.

A portion of this thesis is devoted to the study of models (symbolic representations) of the objects we want to identify; different schemes, some of them already in use, are discussed.

Focusing our attention on the more general problem of identification of general objects when they substantially overlap, we propose some schemes for their recognition, and also analyze some problems that are met. {AD 656-041}

TR-38
Rosenburg, R.C., Kennedy, D.W., Humphrey, R.A.
A LOW-COST OUTPUT TERMINAL FOR TIME-SHARED COMPUTERS
Pages: 25 March 1967 $4.10
Keywords: digital/analog systems, remote terminals, time-sharing
Abstract: This report describes a low-cost remote terminal to provide switch-form output from a time-shared digital computer. The terminal consists of a modified model 35 KSR teletype and a local memory unit. The unit is independent of any 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 automatic set-up and control of an analog computer is described. In this application the results of the analog computation are displayed on an oscilloscope; this makes possible, for example, rapid display of the time response of linear systems, under digital program control. {AD 662-027}

TR-39
Forte, A.
SYNTAX-BASED ANALYTIC READING OF MUSICAL SCORES
Pages: 36 April 1967 $4.40
Keywords: automatic data-structuring, music applications, parsing, syntax-based analysis
Abstract: As part of a larger research project in musical structure, a program has been written which "reads" scores encoded in an input language isomorphic to music notation. The program is believed to be the first of its kind.

From a small number of parsing rules the program derives complex configurations, each of which is associated with a set of reference points in numerical representation of a time-continuum. The logical structure of the program is such that all and only the defined classes of events are represented in the output.

Because the basis of the program is syntactic (in the sense that parsing operations are performed on formal structures in the input string), many extensions and refinements can be made without excessive difficulty. {AD 661-806}

TR-40
Miller, J.R.
ON-LINE ANALYSIS FOR SOCIAL SCIENTISTS
Pages: 25 May 1967 $4.10
Keywords: social science research, expert systems, statistics, time-shared computers
Abstract: A library of computer routines has been compiled to facilitate the analysis of social science research data. Many of these routines are designed to test statistical hypotheses.

All routines are operated on-line and permit conversational interaction between the user and a time-shared computer. Input data are typed directly into the computer through a teletype console. Explicit typing directions and error diagnostics, where appropriate, are printed out by each routine to guide the input process. Analyses are executed immediately, and computed results are printed out in typical publication language.

These routines are designed primarily for social science researchers who do not possess extensive prior training in mathematics, statistics, or computer operations. They provide a rapid, flexible, and immediately accessible method of testing preliminary hypotheses and hunches on small to intermediate amounts of data. They also provide a useful pedagogical tool for training students in practical data analysis.

Detailed instructions for gaining access to the routines are provided in Appendix A of this paper. References to standard statistical texts are also provided so that the user may obtain more detailed information concerning the assumptions underlying each routine and the criteria for selecting them. {AD668-009}

TR-41
Coons, S.A.
SURFACES FOR COMPUTER-AIDED DESIGN OF SPACE FORMS
Pages: 105 June 1967 $6.50
Keywords: computer-aided design, computer graphics, time-sharing
Abstract: The design of airplanes, ships, automobiles, and so-called "sculptured parts" involves the design, delineation, and mathematical description of bounding surfaces. A method is described which makes possible the description of free-form doubly curved surfaces of a very general kind. An extension of these ideas to hyper-surfaces in higher dimensional spaces is also indicated.

This surface technique has been specifically devised for use in the Computer-Aided Design Project at M.I.T., and has already been successfully implemented here and elsewhere. {AD 663-504}

TR-42
Liu, C.L., Chang, G.D., Marks, R.E.
DESIGN AND IMPLEMENTATION OF A TABLE-DRIVEN COMPILER SYSTEM
Pages: 86 July 1967 $5.90
Keywords: compilers, compile generators, syntax-directed compilers, table-driven compilers, time-shared computers
Abstract: Our goal is to provide users of the table-driven compiler system with an environment within which they can freely design and produce their compilers. The primary design criterion is generality so that the users can define a large class of input languages oriented toward any kind of problem-solving purposes, and can also define a large class of object programs to be executed on different computer systems. Therefore, in our system we do not limit the users to specific ways of doing syntactic analysis, or doing storage allocation, or producing binary programs of a specific format for a particular computer system. What we provide are mechanisms that are general enough for whichever way a user desires to build his compiler.

The table-driven compiler system consists of a base program and two fixed higher-level languages - the Table Declaration and Manipulation Language and the Macro Interpretation Language - together with the corresponding translators which generate the control tables according to the user's specification. A third higher-level language - the Syntax Defining Language - and its corresponding translator are also needed. However, their definitions are left to the users for the reason of providing them with greater flexibility in specifying the method of syntactic analysis. The base program is controlled by the control tables to perform the task of translating source programs into object machine codes. It is a general program which is independent of the particular source language being translated as well as the method of translation. The control tables contain an encodement of the syntax of the source language, an encodement of the method of translation and an encodement of the characteristics of the target machine.

In our design, we emphasize the segmentation of the system so that the functions of each section will be clearly defined and be brought out in evidence. The communication problem between the segments is not a difficult one to handle as illustrated in our design. It should also be pointed out that for the generality and flexibility we try to attain, less consideration is placed on efficiency. {AD 668-960}

TR-43
Wilde, D.U.
PROGRAM ANALYSIS BY DIGITAL COMPUTER
Pages: 187 Ph.D. Dissertation/August 1967 $8.95
Keywords: automatic flowcharting, program analysis, time-sharing
Abstract: A comparison of the properties of non-modifying and self-modifying programs leads to the definition of independent and dependent instructions. Because non-modifying programs contain only independent instructions, such programs can be analyzed by a straight forward, two-step analysis procedure. First, the program control flow is detected; second, that control flow is used to determine the program data flow or data processing. However, self-modifying programs can also contain dependent instructions, and the program control flows and data flows exhibit cyclic interaction. This cyclic interaction suggests the use of an iterative or a relaxation analysis technique. The initial step in the relaxation procedure determines a first approximation to data flow. These two steps are repeated until a steady-state condition is reached.

Algorithms for implementing the first iteration are presented. These algorithms are capable of analyzing programs which modify their control and processing instructions during the course of execution. In addition, data structures are described which permit the construction of functional expressions for the data flow or information processing. Finally, actual output flowcharts of self-modifying programs are displayed. {AD 662-224}

TR-44
Gorry, G.A.
A SYSTEM FOR COMPUTER-AIDED DIAGNOSIS
Pages: 244 Ph.D. Dissertation/September 1967 $10.65
Keywords: expert systems, computer-aided diagnosis, medical diagnosis
Abstract: This thesis describes a model diagnostic problem and a computer program designed to deal with this problem. The model diagnostic problem is an abstract problem. A major contention of this thesis, however, is that this problem subsumes the principal features of a number of ostensibly different real diagnostic problems including certain problems of medical diagnosis and the diagnosis of machine failures. A second major contention of this thesis is that strategies for the solution of the model diagnostic problem can be formulated in term sufficiently explicit to permit their incorporation in a computer program.

The diagnostic program was implemented on the time-sharing system at Project Mac. It was applied to two medical problems, the diagnosis of congenital heart disease, and the diagnosis of primary bone tumors. The results obtained here suggest 1) that a computer program can be of considerable value as a diagnostic tool, and 2) that it is quite advantageous for such a program to perform sequential diagnosis as it interacts with the user. {AD 662-665}

TR-45
Leal-Cantu, N.
ON THE SIMULATION OF DYNAMIC SYSTEMS WITH LUMPED PARAMETERS AND TIME DELAYS
Pages: 97 S.M. Thesis/October 1967 $6.25
Keywords: digital computation, digital simulation, dynamic systems, linear dynamic systems, simulation
Abstract: A method is developed for digital simulation of linear time-invariant dynamic systems with lumped parameters and time delays. Ordinarily, such systems can be described by a linear matrix differential-difference equation, which can be transformed to an infinite-dimensional difference equation whose solution is obtained in a recursive way.

As the present method depends on the accuracy of evaluation of the matrix exponential, a simple computational procedure based on the truncation of the infinite series for e^{AT} is described.

In addition, an algorithm is given that ensures that the transient state of an enforced linear time-invariant dynamic system with zero time delay is calculated to a specified accuracy.

Several sample problems are included. {AD 663-502}

TR-46
Alsop, J.W.
A CANONIC TRANSLATOR
Pages: 84 S.B. Thesis/November 1967 $5.85
Keywords: canonic systems, canonic translators, translators
Abstract: An algorithm to recognize and translate sets of character strings specified by canonic systems is presented. the ability of canonic systems to define the context sensitive features of strings and to specify their translation allows the algorithm to recognize and translate real computer languages. It is also applicable in other language systems.

Canonic systems are discussed, and several examples of their use are given. The algorithm is described, and examples of canonic translation are presented using a program which implements it. {AD 663-503}

TR-47
Moses, J.
SYMBOLIC INTEGRATION
Pages: 267 Ph.D. Dissertation/December 1967 $11.35
Keywords: algebraic manipulation, symbolic manipulation, symbolic integration
Abstract: **SIN** and **SOLDIER** are heuristic programs written in LISP which solve symbolic integration problems. **SIN** (**S**ymbolic **IN**tegrator) solves indefinite integration problems at the difficulty approaching those in the larger integral tables. **SIN** contains several more methods than are used in the previous symbolic integration program **SAINT**, and solves most of the problems attempted by **SAINT** in less than one second. **SOLDIER** (**SOL**ution of ordinary **DI**fferential **E**quations **R**outine) solves first order, first degree ordinary differential equations at the level of a good college sophomore and at an average of about five seconds per problem attempted. The difference in philosophy and operation between **SAINT** and **SIN** are described, and suggestions for extending the work presented are made. {AD 662-666}

TR-48
Jones, M.M.
INCREMENTAL SIMULATION ON A TIME-SHARED COMPUTER
Pages: 242 Ph.D. Dissertation/January 1968 $10.60
Keywords: incremental simulation, simulation systems, time-sharing
Abstract: This thesis describes a system which allows simulation models to be built and tested incrementally. It is called OPS-4 and is specifically designed to operate in the environment of the Multics system. I t represents a major expansion and improvement of the OPS-3 system implemented in CTSS and also includes many features adapted from other current simulation systems. The PL-1 language, augmented by many additional statements and new data objects, provides the basis for defining models in OPS-4. A list of desirable features for an incremental simulation system is presented and it is shown how OPS-4 incorporates these features, whereas other current simulation systems satisfy only some of them and are not suitable for use in a time-shared environment. A simplified model of page and segment fault handling in Multics illustrates some of the features OPS-4 provides to allow the user to continuously interact with a model during its construction, testing and running phases. It also illustrates how the user himself may portray portions of a model that are not yet defined. {AD 662-225}

TR-49
Luconi, F.L.
ASYNCHRONOUS COMPUTATIONAL STRUCTURES
Pages: 139 Ph.D. Dissertation/February 1968 $7.50
Keywords: asynchronous systems, computational schema, computational structures
Abstract: The "computational schema" is introduced as a means for describing asynchronous computational structures. This mathematically formulated schema provides for the representation of systems in which several asynchronously communicating processes may proceed concurrently while sharing computational resources. Based on the representation scheme, a theory of asynchronous communication has been developed which allows theorems about the processing characteristics of modular systems to be proved. In particular, deterministic system output behavior is related to a set of conditions on subsystem intercommunication. {AD 677-602}

TR-50
Denning, P.J.
RESOURCE ALLOCATION IN MULTIPROCESS COMPUTER SYSTEMS
Pages: 285 Ph.D. Dissertation/May 1968 $11.90
Keywords: multiprocess computer systems, resource allocation
Abstract: The dynamic allocation for limited processor and main memory resources among members of a user community is investigated as a supply-and-demand problem. The work is divided in four phases. First is the construction of the working set model for program behavior based on locality; a computation's working set is a dynamic measure of this set of favored information. The second phase is the definition and study of properties of system demand. A computation is the basic demand-making entity, placing demands jointly on processor and main memory resources. Its system demand is a pair (processor demand, memory demand). The third phase is the definition and study of the properties of system balance. Computations that demand resources are segregated into two classes - the standby set, which is temporarily denied the use of system resources, and the balance set, which is granted the use of system resources. The system is balanced when the total system demand matches the system capacity. The fourth phase is to apply all these ideas to the design and administration of multiprocess computer systems. {AD675-554}

TR-51
Charniak, E.
CARPS, A PROGRAM WHICH SOLVES CALCULUS WORD PROBLEMS
Pages: 63 S.M. Thesis/July 1968 $5.20
Keywords: calculus problem solving, rate problems
Abstract: A program was written to solve calculus word problems. The program, **CARPS** (**CA**lculus **R**ate **P**roblem **S**olver), is restricted to rate problems. The overall plan of the program is similar to Bobrow's *STUDENT*, the primary difference being the introduction of "structures" as the internal model in **CARPS**. Structures are stored internally as trees. Each structure is designed to hold the information gathered about one object.

A description of **CARPS** is given by working through two problems, one in great detail. Also included is a critical analysis of *STUDENT.* {AD 673-670}

TR-52
Deital, H.M.
ABSENTEE COMPUTATIONS IN A MULTIPLE-ACCESS COMPUTER SYSTEM
Pages: 92 S.M. Thesis/August 1968 $6.10
Keywords: absentee computations, interactive computing, multiple-access computers
Abstract: In multiple-access computer systems, emphasis has been placed upon servicing several interactive users simultaneously. However, computations do not require interaction with the user, and the user may therefore want to run these computations "absentee" (or user not present). A mechanism is presented which provides for the handling of absentee computations in a multiple access computer systems. The design is intended to be implementation-independent; the specifications for each module in the mechanism are presented in terms of the primitive functions which comprise the module. Perhaps the most novel features of the design are the ability for user to switch one of his computations from interactive to absentee (and vice versa), the ability for the system to temporarily suspend and then continue absentee computations to aid in miaintaining an efficient absente-interactive workload on the system, the ability for system administrative personnel to apportion system resources between interactive and absentee computations in order to place emphasis upon a particular mode during certain periods of operation, and the multiple-computation-stream facility of the system which allows the user to attach priorities to his absentee computations by placing the computations in either low-, standard-, or high-priority streams. {AD 684-738}

TR-53
Slutz, D.R.
THE FLOW GRAPH SCHEMATA MODEL OF PARALLEL COMPUTATION
Pages: 256 Ph.D. Dissertation/September 1968 $11.00
Keywords: flow graph schemata, parallel computation, parallel algorithms, determinacy, equivalence, equivalence-preserving transformations
Abstract: Flow graph schemata are introduced as uninterpreted models of parallel algorithms, operating asynchronously and reflecting physical properties inherent to any implementation. Three main topics are investigated: (1) determinacy, (2) equivalence, and (3) equivalence-preserving transformations on the control structure of flow graph schemata. A model is determinate if the results of a computation depend only on the initial values and not on any timing constraints within the model. Equivalence is undecidable in general, but for a large class of determinate flow graph schemata which are in a maximum parallel form, equivalence is shown decidable. In equivalence-preserving transformations, sufficient tested conditions for equivalence are formulated that depend only on the portion of the structure to be transformed.

Current and future computational systems are evaluated in terms of results obtained for flow graph schemata. A number of interesting extensions of the work are suggested. {AD 683-393}

TR-54
Grochow, J.M.
THE GRAPHIC DISPLAY AS AN AID IN THE MONITORING OF A TIME-SHARED COMPUTER SYSTEM
Pages: 74 S.M. Thesis/October 1968 $5.55
Keywords: dynamic monitoring, graphic data display, Multics, time-sharing
Abstract: The problem of dynamic observation of the state of a time-shared computer system is investigated. The Graphical Display Monitoring System was developed as a medium for this experimental work. It is an integrated system for creating graphic displays, dynamically retrieving data from Multics Time-Sharing System supervisor data bases, and on-line viewing of this data via the graphic displays.

On-line and simulated experiments were performed with various members of the Multics staff at Project MAC in an effort to determine what data is most relevant for dynamic monitoring, what display formats are most meaningful, and what sampling rates are most desirable. The particular relevance of using a graphic display as an output medium for the monitoring system is noted.

As a guide to other designers, a generalized description of the principles involved in the design of this on-line, dynamic monitoring device includes special mention of those areas of particular hardware or software system dependence.

Several as yet unsolved problems relating to time-sharing system monitoring, including those of security and data base protection, are discussed. {AD 689-468}

TR-55
Rappaport, R.L.
IMPLEMENTING MULTI-PROCESS PRIMITIVES IN A MULTIPLEXED COMPUTER SYSTEM
Pages: 133 S.M.Thesis/November 1968 $7.30
Keywords: multi-process primitives, process-control primitives, Multics
Abstract: In any computer system primitive functions are needed to control the actions of processes in the system. This thesis discusses a set of six such process control primitives which are sufficient to solve many of the problems involved in parallel processing as well as in the efficient multiplexing of system resources among the many processes in a system. In particular, the thesis documents the work performed in implementing these primitives in a computer system, the *Multics* system, which is being developed at Project MAC of M.I.T. During the course of work that went into the implementation of these primitives, design problems were encountered which caused the overall design of the programs involved to go through two iterations before the performance of these programs was deemed acceptable. The thesis discusses the way design of these programs evolved over the course of work. {AD 689-469}

TR-56
Thornhill, D., Stotz, R.H., Ross, D.T., Ward, J.E.
AN INTEGRATED HARDWARE-SOFTWARE SYSTEM FOR COMPUTER GRAPHICS IN TIME-SHARING
Pages: 147 December 1968 $7.75
Keywords: computer graphics, display software systems
Abstract: This report describes the ESL Display Console and its associated user-oriented software systems developed by the M.I.T. Computer-Aided Design Project with Project MAC. Console facilities include hardware projection of three-dimensional line drawings, automatic light pen tracking, and a flexible set of knob, switch, and push-button inputs. The console is attached to the Project MAC IBM 7094 Compatible Time-Sharing System either directly or through a PDP-7 Computer. Programs of the Display Controller software provide the real-time actions essential to running the display, and communication with the time-sharing supervisor. A companion graphics software system (GRAPHSYS) provides a convenient, high-level, and nearly display-independent interface between the user and the Display Controller. GRAPHSYS procedures allow the user to work with element "picture parts" as well as "subpictures" to which "names" are assigned for identification between user and Controller programs. Software is written mostly in the

machine-independent AED-0 Language of the Project and many of the techniques described are applicable in other contexts. {AD 685-202}

TR-57
Morris, J.H.
LAMBDA CALCULUS MODELS OF PROGRAMMING LANGUAGES
Pages: 131 Ph.D. Dissertation/December 1968 $7.25
Keywords: recursive definitions, type declarations, lambda calculus, programming languages
Abstract: Two aspects of programming languages, recursive definitions and type declarations are analyzed in detail. Church's -calculus is used as a model of a programming language for purposes of the analysis.

The main result on recursion is an analogue to Kleene's first recursion theorem: If A = FA for any λ-expressions A and F, then A is an extension of YF in the sense that if E[YF], any expression containing YF, has a normal form then E[YF] = E[A]. Y is Curry's paradoxical combinator. The result is shown to be invariant for many different versions of Y.

A system of types and type declarations is developed for the l-calculus and its semantic assumptions are identified. The system is shown to be adequate in the sense that it permits a preprocessor to check formulae prior to evaluation to prevent type errors. It is shown that any formula with a valid assignment of types to all its subexpressions must have a normal form. {AD 683-394}

TR-58
Greenbaum, H.J.
A SIMULATOR OF MULTIPLE INTERACTIVE USERS TO DRIVE A TIME-SHARED COMPUTER SYSTEM
Pages: 181 S.M.Thesis/January 1969 $8.75
Keywords: simulators, time-shared computers, Multics, CTSS, performance
Abstract· In the construction and maintenance of a time-shared computer system the need arises for a tool which can provide a controlled, repeatable environment for the purpose of making performance measurements.

This thesis describes the use of a small second computer to simulate the actions of multiple interactive users over individual communication lines. Each simulated user exhibits responses similar to those of a "normal" interactive user. Accordingly, the "Simulator" recognizes and verifies responses transmitted to it by the system being tested. The Simulator also emulates a "think time" corresponding to a normal user's think time between typing lines on the console. Text corresponding to a user's console input, as well as control information regarding think time simulation and verification of responses from the system being tested, are retrieved from prepared scripts which have been pre-sorted on the small computer's magnetic disc unit.

The programming package for the Simulator System has the capability of simulating up to 12 users. For the purpose of this thesis, however, only four users are simulated. The Simulator System is intended to be used to test the M.I.T. CTSS and MULTICS (Time-Shared Computer Systems). However, it is designed to be adaptable for testing most time shared computer systems having serial character oriented input/output over communications lines interfacing with 103A compatible data sets. (AD 686-988)

TR-59
Guzman-Arenas, A.
COMPUTER RECOGNITION OF THREE-DIMENSIONAL OBJECTS IN A VISUAL SCENE
Pages: 287 Ph.D. Dissertation/December 1968 $11.95
Keywords: artificial intelligence, heiristics, pattern recognition, vision, scene analysis
Abstract: Methods are presented 1) to partition or decompose a visual scene into the bodies formint it; 2) to posiiton these bodies in three-dimensional space, by combining two scenes that make a stereoscopic pair; 3) to find the regions or zones of a visual scene that belong to its background; 4) to carry out the isolation of the objects in 1) when the input has inaccuracies. Running computer programs implement the methods, and many examples illustrate their behavior. The input is a two-dimensional line-drawing of the scene, assumed to contain three-dimensional bodies possessing flat faces (polyhedra); some of them may be partially occluded. Suggestions are made for extending the work to curved objects. Some comparisons are made with human visual perception.

The main conclusion is that it is possible to separate a picture or scene into the constituent objects exclusively on the basis of monocular geometric properties (on the basis of pure form); in fact, successful methods are shown. {AD 692-200}

TR-60
Ledgard, H.F.
A FORMAL SYSTEM FOR DEFINING THE SYNTAX AND SEMANTICS OF COMPUTER LANGUAGES
Pages: 203 Ph.D. Dissertation/April 1969 $9.40
Keywords: programming languages, syntax, semantics, lambda calculus
Abstract: The thesis of this dissertation is that formal definitions of the syntax and semantics of computer languages are needed. This dissertation investigates two candidates for formally defining computer languages:

(1) the formalism of canonical systems for defining the syntax of a computer language and its translation into a target language, and

(2) the formalisms of the λ-calculus and extended Markov algorithms as a combined formalism used as the basis of a target language for defining the semantics of a computer language.

Formal definitions of the syntax and semantics of SNOBOL/1 and ALGOL/60 are included as examples of the approach. {AD 689-305}

TR-61
Baecker, R.M.
INTERACTIVE COMPUTER-MEDIATED ANIMATION
Pages: 350 Ph.D. Dissertation/June 1969 $13.85
Keywords: computer animation, computer graphics, interactive graphics
Abstract: The use of interactive computer graphics in the construction of animated visual displays is investigated. In interactive computer-mediated animation, movies are formed from direct console commands, algorithms, free-hand sketches, and real-time actions (such as mimicking a movement or rhythm with a stylus or a push-button). The resulting movies can be immediately viewed and altered.

In picture-driven animation, the animator may sketch and refine (1) static images to be used as components of individual frames of the movie, and (2) static and dynamic images that represent movements and rhythm. These latter pictures drive algorithms to generate dynamic displays. Since each such picture determines critical parameters of a sequence of frames, a single sketch or action controls the dynamic behavior of an entire interval of the movie.

The dissertation also outlines the design of a multi-purpose, open-ended, interactive Animation and Picture Processing Language. APPL is a conversational language which accepts free-hand sketches, real-time actions, and actions and algorithms that control interactive dynamic displays. {AD 690-887}

TR-62
Tillman, C.C.
EPS: AN INTERACTIVE SYSTEM FOR SOLVING ELLIPTIC BOUNDARY-VALUE PROBLEMS WITH FACILITIES FOR DATA MANIPULATION AND GENERAL-PURPOSE COMPUTATION
Pages: 181 June 1969 $8.75
Keywords: computer graphics, interactive problem solving, numerical analysis, elliptical-boundary value problems
Abstract: A user's guide for EPS is presented. EPS solves two-dimensional boundary-value problems for elliptical systems of second-order partial differential equations. It also has general-purpose capabilities which permit the on-line definition and execution of arbitrary numerical procedures.

The guide is primarily concerned with using EPS to solve boundary-value problems. Linear problems of this type that have no free surfaces or undefined parameters can be solved on a one pass basis. Nonlinearities and other complications can be handled by iteration. A finite-difference method is employed which permits the use of irregular lattices, hence the crowding of nodes in sensitive regions.

EPS operates on the IBM 7094 computer of the MIT Compatible Time-Sharing System (CTSS), and exploits to an unusual degree the potential for interactive problem solving that CTSS affords. Input commands resemble statements in various algebraic compiler languages, and can be combined and abbreviated by means of macros. Improper input and other error conditions are handled so as to minimize user inconvenience. Common syntax errors, for example, are corrected automatically by the machine. Output is available in either numerical or graphical form. {AD 692-462}

TR-63
Brackett, J., Hammer, M.M., Thornhill, D.
CASE STUDY IN INTERACTIVE GRAPHICS PROGRAMMING: A CIRCUIT DRAWING AND EDITING PROGRAM FOR USE WITH A STORAGE-TUBE DISPLAY TERMINAL
Pages: 93 October 1969 $6.10
Keywords: computer graphics, computer aided design, display software systems

Abstract: The concepts involved in building and manipulating a data structure through graphical interaction are presented, using the drawing and editing of electrical circuits as a vehicle. The circuit drawing program was designed to operate on an ARDS storage-tube display terminal attached to the M.I.T. Project MAC IBM 7094 Compatible Time-Sharing System. The graphics software system (GRAPHSYS) developed by the M.I.T. Computer-Aided Design Project was used for dealing with all graphical input and output, and the AED Language of the Project was used in programming. AED System packages for building and manipulating complex data structures are described and their use is illustrated in detail. The report includes flow diagrams and complete listings of the sample circuit drawing and editing system. {AD 699-930}

TR-64
Rodrigues, J.E.
A GRAPH MODEL FOR PARALLEL COMPUTATIONS
Pages: 120 Sc.D. Thesis/September 1969 $6.95
Keywords: program graphs, parallel computation, computation models, nodes
Abstract: This report presents a computational model called *program graphs* which makes possible a precise description of parallel computations of arbitrary complexity on non-structured data. In the model, the computation steps are represented by the *nodes* of a *directed graph* whose *links* represent the elements of storage and transmission of *data* and/or *control* information. The activation of the computation represented by a node depends only on the control information residing in each of the links incident into and out of the node. At any given time any number of nodes may be active, and there are no assumptions in the model regarding either the length of time required to perform the computation represented by a node or the length of time required to transmit data or control information from one node to another. Data dependent decisions are incorporated in the model in a novel way which makes a sharp distinction between the local sequencing requirements arising from the data dependency of the computation steps and the global sequencing requirements determined by the logical structure for the algorithm. {AD 697-759}

TR-65
Deremer, F.L.
PRACTICAL TRANSLATORS FOR LR(K) LANGUAGES
Pages: 215 Ph.D. Dissertation/October 1969 $9.80
Keywords: translators, LR(k) languages, compiler writing, translator writing systems, transduction grammars, programming languages, compilers
Abstract: A context-free syntactical translator (CFST) is a machine which defines a translation from one context-free language to another. A transduction grammar is a formal system based on a context-free grammar and it specifies a context-free syntactical translation. A simple suffix transduction grammar based on a context-free grammar which is LR(k) specifies a translation which can be defined by a deterministic push-down automation (DPDA).

A method is presented for automatically constructing CFSTs (DPDAs) from those simple suffix transduction grammars which are based on the LR(k) grammars. The method is developed by first considering grammatical analysis from the string-manipulation viewpoint, then converting the resulting string-manipulation {AD 699-501}

TR-66
Beyer, W.T.
RECOGNITION OF TOPOLOGICAL INVARIANTS BY ITERACTIVE ARRAYS
Pages: 144 Ph.D. Dissertation/October 1969 $7.65
Keywords: topological invariants, iteractive arrays,
The pattern of initial states thus introduced represents a computation based on the input figure. If one waits for a specially designated cell to indicate acceptance of rejection of the figure, the array is said to be working on a recognition problem. If one waits for the array to come to a stable configuration representing an output figure, the array is said to be working on a transformation problem.

Chapter 2 contains a general theory of recognition. Theorems on the amount of time required to perform recognition and on methods of speeding up recognition are presented. Some properties of the classes of recognizable figures are given. Arrays are compared to other types of figure recognition devices. In the last section the class of linear predicates is studied. A linear predicate is a family of figures which can be recognized in time proportional to the perimeter of the figure.

Chapter 3 contains a study of the recognition of some topologically invariant properties of figures. A fundamental transformation of figures is presented and is then used to show that a wide variety of topologically invariant properties form linear predicates including connectivity and maze solvability. Two properties whose linearity is open are discussed.

Chapter 4 contains a brief study of transformation problems. Some general theorems are presented as well as discussions of specific transformations. An optimal solution to the two-dimensional firing squad synchronization problem is also presented in chapter 4.

In addition to the formal results, several open questions are presented and some iterative programming techniques are considered. {AD 699-502}

TR-67
Vanderbilt, D.H.
CONTROLLED INFORMATION SHARING IN A COMPUTER UTILITY
Pages: 172 Ph.D. Dissertation/October 1969 $8.50
Keywords: information sharing, multiple-access computers, controlled information sharing
Abstract: A computer utility is envisioned as a large, multi-access computer system providing its users with the ability to store information and share its use with the ability to store information and share its use with other system users. This thesis considers the nature of information sharing and how a computer utility can provide facilities allowing such sharing to take place in a controlled manner.

From a discussion of the goals of a computer utility, a set of requirements for the facilities of the utility is described. A model is developed which presents a method for structuring information. It is shown that the mechanisms of the model preserve certain structural characteristics of the information, and that these properties can be directly related to the requirements regarding the control of shared information. Extensions of the basic model are described which allow more selective types of control, and which remove some of the limitations of the basic model. {AD 699-503}

TR-68
Selwyn, L.
ECONOMIES OF SCALE IN COMPUTER USE: INITIAL TESTS AND IMPLICATIONS FOR THE COMPUTER UTILITY
Pages: 112 Ph.D. Dissertation/June 1970 $6.70
Keywords: computer utility, economics, time sharing, time-sharing computers, industry installations
Abstract: This study is concerned with the existence of economies of scale in the production of data processing and other computing services, and the possible regulatory and public policy implications of such economies.

The rapid development of the technology of computation since the Second World War has raised many questions as to the supervision of public authorities of the use and progress of this technology. A study was initiated by the Federal Communications Commission in 1966 in an effort to consider that Commission's role in the production and distribution of computing services where the use of communications facilities, supplied by regulated carriers, forms an integral part of the computing system. The present investigation is concerned with the production of computing services per se; the direction that public policy takes will be greatly dependent upon the nature of the production of computing services, and perhaps secondarily upon the interdependence between computer systems and the communications suppliers.

The relative economies of the use of large computing systems have been known for some time, in terms of the relationship between some measure of the quantity of output of a machine and its cost. Indeed, it is demonstrated here that, when one considers, in addition to the cost of the computer hardware itself, the various categories of operating expenses associated with a computer installation, the relative advantages of large facilities become even more significant.

Yet the evidence would seem to indicate that, despite these apparent efficiencies of large systems, the overwhelming majority of installed computers were generally fairly small operations. In an attempt to determine whether actual experience of users was that, all things considered, there were no true economies of large size, an analysis was made of data on nearly 10,000 computers installed at firms in manufacturing industries, using the survival technique, which uses market experience as a basis for studying levels of optimum plant size. The results of this analysis suggested that users did operate computers as if there were significant economies of scale in their use.

None of the evidence, in fact, suggested that even the largest size system available today is the most efficient possible use of "plant" hence, the key implication for the formulation of regulatory policy toward the computer is that such policy should encourage, to the greatest possible extent, the shared use of large systems by those who require computing services. Those barriers that do exist which tend to mitigate such shared use should

be reduced or eliminated. Public utilty status would be indicated only if the costs associated with shared computer use - distribution, software development, system overhead and administration - are less than the potential direct savings resulting from use of large systems. This is at least as much a technological problem as it is regulatory; the future of the computer utility concept will thus be dependent upon the degree to which technology can reduce costs in these categories. (AD 710-001)

TR-69
Gertz, J.L.
HIERARCHICAL ASSOCIATIVE MEMORIES FOR PARALLEL COMPUTATION
Pages: 313 Ph.D. Dissertation/June 1970 $12.70
Keywords: associative memory, parallel computation, memory hierarchies
Abstract: Two current trends in computing, namely the increasing importance of parallelism in computer operations and the concept of programming generality, indicate that new computer systems must employ location-independent addressing. One possible manner of accomplishing this objective involves the use of an associative memory for the computer system, as well as the representation and execution of highly parallel programs in such a memory hierarchy.

The thesis first develops a simple model for the representation of programs which preserves and indicates their inherent parallelism, as well as meshing well with the requirements of programming generality. Then it presents a scheme for representing the resultant objects in an associative memory and develops a corresponding parallel execution algorithm. Next the properties of the memory hierarchy itself are considered. In particular, the use and advantages of modularity are discussed and the method of handling the dynamic allocation of information among the various memory levels is studied in detail. Finally, the program and memory models are linked together in order to develop an overall view of the resultant associative memory computer system. {AD 711-091}

TR-70
Fillat, A.I., Kraning, L.A.
GENERALIZED ORGANIZATION OF LARGE DATA BASES: A SET-THEORETIC APPROACH TO RELATIONS
Pages: 246 S.B. & S.M. Thesis/June 1970 $10.70
Keywords: data base, data structure, data management, relations
Abstract: Problems inherent in representation and manipulation of large data bases are discussed. Data management is considered as the manipulation of relationships among elements of a data base. A detailed analogy introduces concepts embodied in a data management system. Set theory is used to describe a model for data bases, and operations suitable for manipulation of relations are defined. The architecture chosen for an implementation of the model is illustrated, and a representation of data bases is suggested. A particular implementation, the GOLD STAR system, is investigated and evaluated. The framework outlined is meant to provide an environment in which complex data handling problems can be solved with relative ease. GOLD STAR provides the user with tools sufficient for manipulation of arbitrary complex data bases; these provisions are presented in the form of an extremely simple interface. {AD 711-060}

TR-71
Fiasconaro, J.G.
A COMPUTER-CONTROLLED GRAPHICAL DISPLAY PROCESSOR
Pages: 69 S.M. Thesis/June 1970 $5.40
Keywords: graphic display, display processor
Abstract: A cathode-ray tube, (CRT), is frequently employed to display text and drawings generated by a digital computer. Unfortunately, all of the commercially available CRT display systems are either very expensive or have limited dynamic capability resulting from the use of some form of storage-type CRT. A need exists to develop a low-cost, relatively sophisticated display processor that can be used with a standard CRT monitor to display computer-generated pictures.

This thesis describes the design, construction and operation of such a display processor. The design was based on the following desired characteristics. The display processor should be able to intensify any one of the 1024 X 1024 raster points of the 10 X 10 inch display area. It should be able to display any one of the 96 printable ASCII characters. And finally, it should provide two character sizes and eight intensity lees.

The prototype that was built based on the above characteristics was found to work quite well. The total parts cost for the display processor was only about $3,000. Thus it can be concluded that it was possible to achieve the goal of developing a useful low-cost display processor for a CRT display system. {AD 710-479}

TR-72
Patil, S.S.
COORDINATION OF ASYNCHRONOUS EVENTS
Pages: 234 Sc.D. Thesis/June 1970 $10.35
Keywords: multi-processing, parallel-processing, digital logic, computer systems, asynchronous systems, computational schema, resource management, coordination of events
Abstract: The way activity in a system proceeds is that events occur as a result of some conditions and lead to some new conditions which make other events possible. Often it is necessary to coordinate such events to ensure proper behavior. Coordination nets for representing such coordinations and physically realizable structures are modular and can be mechanically derived from the coordination nets. Coordinations involved in concurrent management of resources are also discussed. {AD 711-763}

TR-73
Griffith, A.K.
COMPUTER RECOGNITION OF PRISMATIC SOLIDS
Pages: 181 Ph.D. Dissertation/August 1970 $8.75
Keywords: pattern recognition, artificial intelligence, statistical decision theory, picture processing, perception, feature recognition, picture abstraction, machine aided cognition, vision, robotics
Abstract: An investigation is made into the problem of constructing a model of the appearance to an optical input device of scenes consisting of plane-faced geometric solids. The goal is to study algorithms which find the real straight edges in the scenes, taking into account smooth variations in intensity over the faces of the solids, blurring of edges and noise. A general mathematical analysis is made of optimal methods for identifying the edge lines in figures, given a raster of intensities covering the entire field of view. There is given in addition a suboptimal statistical decision procedure, based on the model, for the identification of a line within a narrow band on the field of view given an array of intensities from within the band. A computer program has been written and extensively tested which implements this procedure and extracts lines from real scenes. Other programs were written which judge the completeness of extracted sets of lines, and propose and text for additional lines which had escaped initial detection. The performance of these programs is discussed in relation to the theory derived from the model, and with regard to their use of global information in detecting and proposing lines. {AD 712-069}

TR-74
Edelberg, M.
INTEGRAL CONVEX POLYHEDRA AND AN APPROACH TO INTEGRALIZATION
Pages: 170 Ph.D. Dissertation/August 1970 $8.45
Keywords: integral convex polyhedra, integral convex hulls, integer linear programming, combinatorial optimization
Abstract: Many combinatorial optimization problems may be formulated as integer linear programming problems -- that is, problems of the form: given a convex polyhedron P contained in the non-negative orthant of ndimensional space, find an integer point in P which maximizes (or minimizes) a given linear objective function. Well known linear programming methods would suffice to solve such a problem if:

(i) P is an integral convex polyhedron, or

(ii) P is transformed into the integral convex polyhedron that is the convex hull of the set of integer points in P, a process which is called integralization.

This thesis provides some theoretical results concerning integral convex polyhedra and the process of integralization. Necessary and sufficient conditions for a convex polyhedron P to have the integral property are derived in terms of the system of linear inequalities defining P. A number-theoretic method for integralizing two-dimensional convex polyhedra is developed which makes use of a generalization of the division theorem for integers. The method is applicable to a restricted class of higher dimensional polyhedra as well. {AD 712-070}

TR-75
Hebalkar, P.G.
DEADLOCK-FREE SHARING OF RESOURCES IN ASYNCHRONOUS SYSTEMS
Pages: 185 Sc.D. Thesis/September 1970 $8.90
Keywords: resource allocation, deadlock, asynchronous systems, project scheduling
Abstract: Whenever resources are shared among several activities that hoard resources, the activities can attain a state of deadlock in which, for lack of resources, none of the activities can proceed. Deadlocks can be

prevented by coordination of the sharing. Efficient running of the activities under such coordination requires knowledge of the patterns of use of resources by the activities.

This thesis presents a study of deadlock prevention in systems in which a knowledge of the usage of resources by the activities during several phases of steady resource usage is available. A representation called a demand graph is presented and used for the study of deadlocks. The model is a general one and encompasses systems in which the activities themselves consist of more than one sequence of phases and are not necessarily independent of each other. The analysis is applicable to computer systems as well as systems in the realm of operations research. {AD 713-139}

TR-76
Winston, P.H.
LEARNING STRUCTURAL DESCRIPTION FROM EXAMPLES
Pages: 265 Ph.D. Dissertation/September 1970 $11.30
Keywords: artificial intelligence, learning, vision, machine perception, network memory, matching, recognition
Abstract: The research here described centers on how a machine can recognize concepts and learn concepts to be recognized. Explanations are found in computer programs that build and manipulate abstract descriptions of scenes such as those children construct from toy blocks. One program uses sample scenes to create models of simple configurations like the three-brick arch. Another uses the resulting models in making identifications. Throughout, emphasis is given to the importance of using good descriptions when exploring how machines can come to perceive and understand the visual environment. {AD 713-988}

TR-77
Haggerty, J.P.
COMPLEXITY MEASURES FOR LANGUAGE RECOGNITION BY CANONIC SYSTEMS
Pages: 46 S.M. Thesis/October 1970 $4.70
Keywords: formal systems, logical systems, complexity measures, canonic systems
Abstract: A canonic system C is a specification of a recursively enumerable set, such as a set of strings over a finite alphabet. From this description C, it is possible to generate a system C_m, called a proof measure function, which is an indication of the complexity of the language defined. For certain simple but important classes of canonic systems, algebraic bounds on these functions can be derived from the structure of the system. Another transformation on C produces a system C^{-1}, which characterizes the recognition of strings generated by C. A relationship exists between the measure functions of C and C^{-1}, thus relating the complexity of the recognition procedure to that of the language description. {AD 715-134}

TR-78
Madnick, S.E.
DESIGN STRATEGIES FOR FILE SYSTEMS
Pages: 106 S.M. Thesis/October 1970 $6.50
Keywords: file systems, operating systems, modularity, virtual memory, data management
Abstract: This thesis describes a methodology for the analysis and synthesis of modern general purpose file systems. The two basic concepts developed are (1) establishment of a uniform representation of a file's structure in the form of virtual memory or segmentation and (2) determination of a hierarchy of logical transformations within a file system. These concepts are used together to form a strictly hierarchical organization (after Dijkstra) such that each transformation can be described as a function of its lower neighboring transformation. In a sense, the complex file system is built up by the composition of simple functional transformations. To illustrate the specifics of the design process, a file system is synthesized for an environment including a multi-computer network, structured file directories, and removable volumes. {AD 714-269}

TR-79
Horn, B.K.P.
SHAPE FROM SHADING: A METHOD FOR OBTAINING THE SHAPE OF A SMOOTH OPAQUE OBJECT FROM ONE VIEW
Pages: 196 Ph.D. Dissertation/November 1970 $9.20
Keywords: artificial intelligence, image processing, depth cues, vision
Abstract: A method will be described for finding the shape of a smooth opaque object from a monocular image, given a knowledge of the surface photometry, the position of the light-source and certain auxiliary information to resolve ambiguities. This method is complementary to the use of stereoscopy which relies on matching up sharp detail and will fail on smooth objects. Until now the image processing of single views has been restricted to objects which can meaningfully be considered two-dimensional or bounded by plane surfaces.

It is possible to derive a first-order non-linear partial differential equation in two unknowns relating the intensity at the image points to the shape of the object. This equation can be solved by means of an equivalent set of five ordinary differential equations. A curve traced out by solving this set of equations for one set of starting values is called a characteristic strip. Starting one of these strips from each point on some initial curve will produce the whole solution surface. The initial curves can usually be constructed around so-called singular points.

A number of applications of this method will be discussed including one to lunar topography and one to the scanning electron microscope. In both of these cases great simplifications occur in the equations. A note on polyhedra follows and a quantitative theory of facial make-up is touched upon.

An implementation of some of these ideas on the PDP-6 computer with its attached image-dissector camera at the Artificial Intelligence Laboratory will be described, and also a nose-recognition program. {AD 717-336}

TR-80
Clark, D.D., Graham, R.M., Saltzer, J.H., Schroeder, M.D.
THE CLASSROOM INFORMATION AND COMPUTING SERVICE
Pages: 278 January 1971 $11.65
Keywords: time-shared computers, multiplexed computers, classroom computer utility model, Multics, computation utility systems, multiple-access computers, operating systems
Abstract: This report describes the Classroom Information and Computing Service (Clics), a pedagogical computer-based information system that is used as a case study in the subject"Information Systems" in the Department of Electrical Engineering at M.I.T. Clics is an abstraction of the Multiplexed Information and Computing Service (Multics) that is being implemented by Project MAC at M.I.T. As such, it is an example of a computer utility. Clics is derived from Multics by a combination of simplifying the mechanisms of Multics and removing some of its more exotic features; and embodies research into ways to simplify the mechanisms of Multics without sacrificing service objectives.

This report is a specification of the hardware, control programs, and system implementation language of the Clics system, as developed to date. The system is specified in sufficient detail for students to develop a structural as well as a functional understanding of its operation and mechanisms. As the primary case study for an undergraduate subject, Clics provides specific examples of the complexities in a general purpose information system, and methods of coping with them.

TR-81
Banks, E.R.
INFORMATION PROCESSING AND TRANSMISSION IN CELLULAR AUTOMATA
Pages: 100 Ph.D. Dissertation/January 1971 $6.35
Keywords: cellular automata, iterative arrays, universal computers
Abstract: A cellular automaton is an iterative array of very simple identical information processing machines called cells. Each cell can communicate with neighboring cells. At discrete moments of time the cells can change from one state to another as a function of the states of the cell and its neighbors. Thus on a global basis, the collection of cells is characterized by some type of behavior.

The goal of this investigation was to determine just how simple the individual cells could be while the global behavior achieved some specified criterion of complexity -- usually the ability to perform a computation or to reproduce some pattern.

The chief result described in this thesis is that an array of identical square cells (in two dimensions), each cell of which communicates directly with only its four nearest edge neighbors and each of which can exist in only two states, can perform any computation. This computation proceeds in a straight forward way.

A configuration is a specification of the states of all the cells in some area of the iterative array. Another result described in this thesis is the existence of a self-reproducing configuration in an array of four-state cells, a reduction of four states from the previously known eight-state case.

The technique of information processing in cellular arrays involves the synthesis of some basic components. Then the desired behaviors are obtained by the interconnection of these components. A chapter on components describes some sets of basic components.

Possible applications of the results of this investigation, descriptions of some interesting phenomena (for vanishingly small cells), and suggestions for further study are given later. {AD 717-951}

TR-82
Krakauer, L.J.
COMPUTER ANALYSIS OF VISUAL PROPERTIES OF CURVED OBJECTS
Pages: 125 Ph.D. Dissertation/May 1971 $7.10
Keywords: artificial intelligence, pattern recognition, thinking machines, vision
Abstract: A method is presented for the visual analysis of objects by computer. It is particularly well suited for opaque objects with smoothly curved surfaces. The method extracts information about the object's surface properties, including measures of its specularity, texture, and regularity. It also aids in determining the object's shape.

The application of this method to a simple recognition task, (the recognition of fruit) is discussed. The results on a more complex smoothly curved object, a human face, are also considered. {AD 723-647}

TR-83
Lewin, D.
IN-PROCESS MANUFACTURING QUALITY CONTROL
Pages: 560 Ph.D. Dissertation/January 1971 $20.10
Keywords: adaptive control, design of simulation experiments, modeling, economics of manufacturing, quality control, optimization of simulation models,resource allocation, simulation as an adjunct to analysis, robustness analysis
Abstract: The thesis develops a methodology for designing plans for the allocation of in-process inspection effort. The focus of the thesis is on constructing operating rules for the allocation of inspection effort along a production line in which inspection and repair are integral parts of that line. The essential feature of such operating rules is their adaptability, i.e., their capacity to detect and respond to changes in the quality levels at various parts of the manufacturing process. The basic methodology is an application of micro-economic analysis to the production process. The major features restricting the problem setting needed for the proposed methodology to be applicable are the following: 1) discrete production units and production sub-processes; 2) identifiability of a finite set of independent attribute defects (or surrogate defects); 3) no information available to monitor the current state of the production process by any means other than inspection of the product; 4) nondestructive inspection, and 5) repairable defects. {AD 720-098}

TR-84
Winograd, T.
PROCEDURES AS A REPRESENTATION FOR DATA IN A COMPUTER PROGRAM FOR UNDERSTANDING NATURAL LANGUAGE
Pages: 462 Ph.D. Dissertation/February 1971 $17.10
Keywords: natural language, theorem proving, PLANNER, linguistics, semantics, parsing, machine understanding
Abstract: This paper describes a system for the computer understanding of English. The system answers questions, executes commands, and accepts information in normal English dialog. It uses semantic information and context to understand discourse and to disambiguate sentences. It combines a complete syntactic analysis of each sentence with a "heuristic understander" which uses different kinds of information about a sentence, other parts of the discourse, and general information about the world in deciding what the sentence means.

It is based on the belief that a computer cannot deal reasonably with language unless it can "understand" the subject it is discusing. The program is given a detailed model of the knowledge needed by a simple robot having only a hand and an eye. We can give it instructions to manipualte toy objects, interrogate it about the scend and give it informatio it will use in deduction. In addition to knowing the properties of toy objects, the program has a simple model of its own mentality. It can remember and discuss its plans and actions as well as carry them out. It enters into a dialog with a person, responding to English sentences with actions and English replies, and asking for clarification when its heuristic programs cannot understand a sentence through use of context and physical knowledge.

In the programs, syntax, semantics and inference are integrated in a "vertical" system in which each part is constantly communicating with the others. We have explored several techniques for integrating the large bodies of complex knowledge needed to understand language. We use Systemic Grammar, a type of syntactic analysis which is designed to deal with semantics. Rather than concentrating on the exact form of rules for the shapes of linguistic constituents, it is structured around choices for conveying meaning. It abstracts the relevant features of the linguistic structures which are important for interpreting their meaning.

We represent many kinds of knowledge in the form of procedures rather than tables of rules or lists of patterns. By developing special procedural languages for grammar, semantics, and deductive logic, we gain the flexibility and power of programming languages while retaining the regularity and understandability of simpler rule forms. Each piece of knowledge can be a procedure, and can call on any other piece of knowledge in the system. {AD 721-399}

TR-85
Miller, P.L.
AUTOMATIC CREATION OF A CODE GENERATOR FROM A MACHINE DESCIPTION
Pages: 98 E.E. Thesis/May 1971 $6.25
Keywords: compilers, translator writing systems, machine code generation, programming languages
Abstract: This paper studies some of the problems involved in attaining machine independence for a code generator, similar to the language independence and the token independence attained by automatic parsing and automatic lexical systems. In particular, the paper examines the logic involved in two areas of code generation: computation and data reference. It presents models embodying the logic of each area and demonstrates how the models can be filled out by descriptive information about a particular machine. The paper also describes how the models can be incorporated into a descriptive macro code generating system (DMACS) to be used as a tool by a language implementor in creating a machine independent code generator, which can be made machine-directed by a suitable description of a particular machine. {AD 724-730}
TR-86
Schell, R.R
DYNAMIC RECONFIGURATION IN A MODULAR COMPUTER SYSTEM
Pages· 189 P.D. Dissertation/June 1971 $9.00
Keywords: modular computer systems, computer utility, Multics, dynamic reconfiguration, operating systems
Abstract: This thesis presents an orderly design approach for dynamically changing the configuration of constituent physical units in a modular computer system. Dynamic reconfiguration contributes to high system availability by allowing preventative maintenance, development of new operating systems, and changes in system capacity on a non-interference basis. The design presented includes the operating system primitives and hardware architecture for adding and removing any (primary or secondary) storage module and associated processing modules while the system is running. Reconfiguration is externally initiated by a simple request from a human operator and is accomplished automatically without disruption to users of the system. This design allows the modules in an installation to be partitioned into separate non-interfering systems. The viability of the design approach has been demonstrated by employing it for a practical implementation of processor and primary memory dynamic reconfiguration in the Multics system at M.I.T. {AD 725-859}

TR-87
Thomas, R.H.
A MODEL FOR PROCESS REPRESENTATION AND SYNTHESIS
Pages: 267 Ph.D. Dissertation/June 1971 $11.35
Keywords: cooperating sequential processes, extensible programming languages, programming languages, process notion, language extension, language definition, control structures, interrupts
Abstract: This dissertation investigates the problem of representing groups of loosely connected processes and develops a model for process representation useful for synthesizing complex patterns of process behavior. There are three parts to the dissertation. The first part isolates the concepts which for the basis for the process representation model by focusing on questions such as: What is a process; What is an event; Should one process be able to restrict the capabilities of another? The second part develops a model for process representation which captures the concepts and intuitions developed in the first part. The model presented is able to describe both the internal structure of individual processes and the "interface" structure between interacting processes. Much of the models descriptive power derives from its use of the notion of process state as a vehicle for relating the internal and external aspects of process behavior. The third part demonstrates by example that the model for process representation is a useful one for synthesizing process behavior patterns. In it the model is used to define a variety of interesting process behavior patterns. The dissertation closes by suggesting how the model could be used as a semantic base for a very potent language extension facility. {AD 726-049}

TR-88
Welch, T.A.
BOUNDS ON INFORMATION RETRIEVAL EFFICIENCY IN STATIC FILE STRUCTURES
Pages: 164 Ph.D. Dissertation/June 1971 $8.25
Keywords: information theory, file organization, information retrieval, classification, directories, relevance, libraries
Abstract: This research addresses the problem of file organization for efficient information retrieval when each file item may be accessed through any one of a large number of identification keys. The emphasis is on library problems, namely large, low-update, directory oriented files, but other types of files are discussed. The model used introduces the concept of an ideal directory against which all imperfect real implementations (catalogs) can be compared. The use of an ideal reference point serves to separate language interpretation problems from information organization problems, and permits concentration on the latter. The model includes a probabilistic description of file usage, developed to give precise definition to the range of user requirements. The analysis employs mathematical tools and techniques developed for information theory, such as the entropy measure and the concept of an ensemble of possible file items.

The principal analysis variable is item relevance, the probability that a file item accessed is actually useful, which is a measure of retrieval efficiency. An upper bound on average relevance is derived, and is found to give useful results in two areas. First, it shows that retrieval efficiency is determined primarily by catalog size (amount of information stored) and user question statistics, with only second-order effects due to type of catalog data and file structure used. Second, it is used to evaluate various indexing procedures proposed for libraries and to suggest improved experimental procedures in this field. {AD 725-429}

TR-89
Owens, R.C.
PRIMARY ACCESS CONTROL IN LARGE-SCALE TIME-SHARED DECISION SYSTEMS
Pages: 92 S.M. Thesis/July 1971 $6.10
Keywords: access control, privacy, controlled information sharing
Abstract: Four primary dimensions of the access control problem are identified. They are: 1) the physical level at which to apply control; 2) the fineness of distinction applied to the term "access"; 3) the meaning of the term "user identification"; and 4) the degree of sophistication employed in automatically assigning sensitivity descriptions to derived data sets.

Within the context of MacAIMS, the Project MAC Advanced Interactive Management System, the design of an access control system is presented which takes positions along these dimensions appropriate for controlling access in a Management Decision System. Particular attention is given to computing access characteristics of new data sets derived from existing sets.

In addition, several classes of problems which the system cannot solve are presented. {AD 728-036}

TR-90
Lester, B.P.
COST ANALYSIS OF DEBUGGING SYSTEMS
Pages: 112 S.B. & S.M. Thesis/September 1971 $6.70
Keywords: programming linguistics, interpreters, interactive computing, semantic definition, cost analysis, debugging, time-sharing
Abstract: A general method is presented for performing cost analysis of interactive debugging systems. The method is based on an abstract model of program execution. This model is derived from the interpreter used in the Vienna method of semantic definition of PL/I. A brief discussion of the overall operation and significance of the Vienna interpreter is included.

Four assumptions are made which allow execution times to be calculated for algorithms of the Vienna interpreter. A notion of absolute cost is developed which requires the use of these execution times for cost analysis of features of debugging systems. A set of eight interactive debugging operations is thoroughly analyzed using the method of cost analysis. Some overall conclusions are drawn about the relative costs of various types of debugging operations and some suggestions are made for minimal cost debugging system design. {AD 730-521}

TR-91
Smoliar, S.W.
A PARALLEL PROCESSING MODEL OF MUSICAL STRUCTURES
Pages: 275 Ph.D. Dissertation/September 1971 $11.60
Keywords: EUTERPE, musical synthesis, computer music, sound-synthesis
Abstract: EUTERPE is a real-time computer system for the modeling of musical structures. It provides a formalism wherein familiar concepts of musical analysis may be readily expressed. This is verified by its application to the analysis of a wide variety of conventional forms of music, Gregorian chant, Medieval polyphony, Back countgerpoint, and sonata form. It may be of further assistance in the real-time experiments in various techniques of thematic development. Finally, the system is endowed with sound-synthesis apparatus with which the user may prepare tapes for musical performances. {AD 731-690}

TR-92
Wang, P.S.
EVALUATION OF DEFINITE INTEGRALS BY SYMBOLIC MANIPULATION
Pages: 182 Ph.D. Dissertation/October 1971 $8.80
Keywords: algebraic manipulation, mathematics, MACSYMA
Abstract: A heuristic computer program for the evaluation of real definite integrals of elementary functions is described. This program, called WANDERER (**WAN**g's **DE**finite integ**R**al Evaluato**R**), evaluates many proper and improper integrals. The improper integrals may have a finite or infinite range of integration. Evaluation by contour integration and residue theory is among the methods used. A program called DELIMITER (**DE**finite **LIMIT** Evaluato**R**) is used for the limit computations needed in evaluating some definite integrals. DELIMITER is a heuristic program written for computing limits of real or complex analytic functions. For real functions of a real variable, one-sided as well as two-sided limits can be computed. WANDERER and DELIMITER have been implemented in the MACSYMA system, a symbolic and algebraic manipulation system being developed at Project Mac, MIT. A typical problem in applied mathematics, namely asymptotic analysis of a definite integral, is solved using MACSYMA to demonstrate the usefulness of such a system and the facilities provided by WANDERER. {AD 732-005}

TR-93
Greif, I.
INDUCTION IN PROOFS ABOUT PROGRAMS
Pages: 60 S.M. Thesis/February 1972 $5.15
Keywords: recursion induction, proofs about programs, automatic programming
Abstract: Four methods for proving equivalence of programs y induction are described and compared. They are recursion induction, structural induction, mu-rule induction, and truncation induction. McCarthy's formalism for conditional expressions as function definitions is used and reinterpreted in view of Park's work on results in lattice theory as related to proofs about programs. The possible application of this work to automatic program verification is commented upon. {AD 737-701}

TR-94
Hack, M.
ANALYSIS OF PRODUCTION SCHEMATA BY PETRI NETS
Pages: 119 S.M. Thesis/February 1972 $6.90
Keywords: Petri nets, concurrent systems, production schemata
Abstract: Petri nets provide a powerful graphical tool for representing and analyzing complex concurrent systems. Properties such as hang-up freeness, determinacy, conflict, concurrency and dependency, can be represented and studied. The precise relationship between structural and behavioral properties, and between local and global properties is not well-understood for the most general class of Petri Nets. This thesis presents such results for a restricted class of Petri Nets called Free Choice Petri Nets, and for a corresponding class of systems called Production Schemata. Results on structural constraints guaranteeing global operation, and decompositions of complex systems into meaningful parts, are also presented. {AD 740-320}

TR-95
Fateman, R.J.
ESSAYS IN ALGEBRAIC SIMPLIFICATION
Pages: 190 Ph.D. Dissertation/April 1972 $9.05
Keywords: mathematics, algebraic manipulation, canonical simplification, MACSYMA
Abstract: This thesis consists of essays on several aspects of the problem of algebraic simplification by computer. Since simplification is at the core of most algebraic manipulations, efficient and effective simplification procedures are essential to building useful computer systems for non-numerical mathematics. Efficiency is attained through carefully designed and engineered algorithms, heuristics, and data types, while effectiveness is assured through theoretical considerations.

Chapter 1 is an introduction to the field of algebraic manipulation, and serves to place the following chapters in perspective.

Chapter 2 reports on an original design for, and programming implementation of, a pattern matching system intended to recognize non-obvious occurrences of patterns within algebraic expressions. A user of such a system can "teach" the computer new simplification rules.

Chapter 3 reports on new applications of standard mathematical algorithms used for canonical simplifications of rational expressions. These applications, in combinations, allow a computer system to contain a fair amount of expertise in several areas of algebraic manipulation.

Chapter 4 reports on a new, practical, canonical simplification algorithm for radical expressions (i.e. algebraic expressions including roots of polynomials). The effectiveness of the procedure is assured through proofs of appropriate properties of these simplified expressions.

Chapter 5 is a brief summary and a discussion of potential research areas.

Two appendices describe MACSYMA, a computer system for symbolic manipulation, an effort of some dozen researchers (including the author) which has served as the vehicle for this work. {AD 740-132}

TR-96
Manning, F.
AUTONOMOUS, SYNCHRONOUS COUNTERS CONSTRUCTED ONLY OF J-K FLIP-FLOPS
Pages: 64 S.M. Thesis/May 1972 $5.25
Keywords: synchronous counters, J-K flip-flops
Abstract: This report describes research into some properties of autonomous, synchronous counters constructed with only the simplest form of J-K Flip-Flop.

The research revolved around a system with a special-purpose digital machine and a general-purpose computer. The special-purpose machine searched through all the possible counters constructed of five or fewer J-K Flip-Flops for all counters with a period equal to that specified by the input to the system. The descriptions of the counters found were transmitted by the special-purpose machine to the computer. Software analyzed this output data for various attributes, such as counters that cycle the same way independent of their starting state. Useful information for designers of digital machines, several proofs, and some insight into these counters resulted from this analysis. {AD 744-030}

TR-97
Vilfan, B.
THE COMPLEXITY OF FINITE FUNCTIONS
Pages: 117 Ph.D. dissertation/March 1972 $6.85
Keywords: finite functions, cyclic perceptrons
Abstract: Lower bounds on the length of formulas for finite functions are obtained from a generalization of a theorem of Specker. Let f: $\{0,1,...,d\text{-}1\}^n \rightarrow \{0,1,...,d\text{-}1\}$ be a function which can be represented by a formula of length $\leq c \cdot n$. For any m, if n is sufficiently large, there is a restriction f': $\{0,1,...,d\text{-}1\}^m \rightarrow \{0,1,...,d\text{-}1\}$ of f which is representable by a special class of formulas called homogeneous e-complexes. By showing that certain functions do not have restrictions representable by homogeneous e-complexes, we are able to conclude that the length of formulas representing the mod p sum, p > d, or the connectedness of a pattern on a discrete retina cannot be bounded by a linear function of the number of variables in the formula. .br Also considered are perceptrons over finite fields (cyclic perceptrons). It is shown that cyclic perceptrons of bounded order cannot represent the g eometric predicate connectivity. An interesting aspect of this is that one proof of the corresponding result for bounded order perceptrons over the rationals rests on the inability of the latter to represent the parity function. However, the parity function requires order 1 if the field has characteristic 2; thus, this proof breaks down in the case of cyclic perceptrons. Another geometric predicate that cannot be represented by bounded order cyclic perceptrons is Euler number equals k (for an arbitrary k). However, this predicate can be represented by bounded order perceptrons over the rationals. It must be noted, however, that our proofs are different and much simpler than the corresponding proofs derived by Minsky and Papert for perceptrons over the rationals.

Finally, we investigate k-pattern spectra of a discrete retina. This is the 2^{k^2}-tuple, each component of which corresponds to the number of times a particular kxk pattern occurs on the retina. It is shown that the only topological predicates that can be determined from k-pattern spectra of discrete figures are functions of the Euler number of the figure. {AD 739-678}

TR-98
Stockmeyer, L.J.
BOUNDS ON POLYNOMIAL EVALUATION ALGORITHMS
Pages: 54 S.M. Thesis/April 1972 $4.95
Keywords: mathematics, polynomial evaluation algorithm
Abstract: The purpose of this work is to investigate the number of arithmetic operations required by algorithms which evaluate polynomials. Previous results show that a polynomial of degree n requires at least n/2 multiplication/divisions and at least n addition/subtractions for its evaluation if the coefficients of the polynomial are suitably independent irrational numbers. However, the coefficients of any polynomial that would be evaluated in practice are represented only to a finite accuracy and are therefore rational numbers. The above results are extended to show that the same lower bounds hold for almost all rational polynomials if the polynomial is being evaluated efficiently. Another lower bound result is given that shows that almost all rational polynomials of degree n require at least (sq. rt. n) multiplication/divisions for their evaluation by any algorithm, efficient or not.

Several algorithms are presented which can in theory evaluate any rational polynomial using O(sq. rt.(n)) multiplications and many additions. While of no practical use for rational polynomials in general, these algorithms do turn out to give methods for evaluating a polynomial at a matrix argument which are more efficient than previous methods. {AD 740-328}

TR-99
Lynch, N.A.
RELATIVIZATION OF THE THEORY OF COMPUTATIONAL COMPLEXITY
Pages: 125 Ph.D. Dissertation/June 1972 $7.10
Keywords: computational complexity, Turing machine, recursive functions, relatedness
Abstract: Blum's machine-independent treatment of the complexity of partial recursive functions is extended to relative algorithms (as represented by Turing machines with oracles). We prove relativizations of several results of Blum complexity theory, such as the compression theorem. A recursive relatedness theorem is proved, showing that any two relative complexity measures are related by a fixed recursive function. This theorem allows us to obtain proofs of results for *all* measures from proofs for a particular measure.

We study complexity-determined reducibilities, the parallel notion to complexity classes for the relativized case. Truth-table and primitive recursive reducibilities are reducibilities of this type, while other commonly-studied reducibilities are not.

We formalize the concept of a set helping the computation of function (by causing a saving in resource when used as an oracle in the computation of the function). Basic properties of the "helping" relation are proved, including non-transitivity and bounds on the amount of help certain sets can provide.

Several independence results (results about sets that don't help each other's computation) are proved; they are subrecursive analogs to degrees-of-unsolvability theorems, with similar proofs using diagonalization and priority arguments. In particular, we discuss the existence of a "universally-helped set," obtaining partial results in both directions. The deepest result is a finite-injury priority argument (without an apparent recursive bound on the number of injuries) which produces sets preserving an arbitrary lower bound on the complexity of a set.

Our methods of proof include proof for a simple measure (e.g. space) and appeal to recursive relatedness, diagonalization and priority techniques, and heavy use of arguments about the domain of convergence of partial recursive functions in order to define total recursive functions. {AD744-032}

TR-100
Mandl, R.
FURTHER RESULTS ON HIERARCHIES OF CANONIC SYSTEMS
Pages: 84 S.M.Thesis/June 1972 $5.85
Keywords: simple canonic systems, general canonic systems, heterogeneous, hierarchy, non-cross referencing, non-inserting
Abstract: This thesis outlines a new way of presenting the theory of canonic systems, including a distinction (for methodic reasons) between simple canonic systems and general canonic systems, and probes a series of results on hierarchies of canonic systems. After a brief summary of Doyle's results on a partial hierarchy of canonic systems, a new hierarchy is developed (Chapter II) which relates the general canonic systems not only to all 4 types of formal grammars defined by Chomsky but also to any class of formal grammars definable in terms of productions. It is also shown

(Chapter iii) that all attempts to define a mathematical system which exactly corresponds to the recursive sets are necessarily fruitless. Doyle's work on how to define "non-contracting canonic systems with predicates of degree 2" (NCST) is continued, arriving at a workable definition which permits us to prove [NCST] = [Type 1] (Chapter IV), a conjecture put forth at the III'd Princeton Conference on Information Sciences and Systems. This results transforms Doyle's hierarchy from "the union of two half-hierarchies and a dangling term (the NCST)" into a complete hierarchy of canonic systems (all 4 types represented). However, this hierarchy is heterogeneous: canonic systems corresponding to grammars of types 3 and 2 use only predicated of degree 1, while canonic systems corresponding to grammars of types 1 and 0 use also predicates of degree 2; moreover, for all types of grammars except for context-sensitive grammars the canonic systems turned out to be simple. Schematically, the form of this hierarchy may be summarized as

S1 S1 G2 S2 (for types 3, 2, 1, 0).

We first show (Chapter V) how to get a hierarchy of simple canonic systems,

S1 S1 S2 S2,

using as base Doyle's hierarchy, and then transform it into

S1 S1 S2 S1.

Since this hierarchy does not seem to lend itself to further "homogenization", we shall use the hierarchy of Chapter II to obtain a hierarchy of simple canonic systems with predicates of degree 1:

S1 S1 S1 S1.

Several new classes of canonic systems (non-crossreferencing, non-inserting, and pure canonic systems) are introduced in Chapter VI, where their properties are explored, and a classification schema and several hierarchies are developed. {AD 744-206}

TR-101
Dennis, J.B.
ON THE DESIGN AND SPECIFICATION OF A COMMON BASE LANGUAGE
Pages: 46 June 1972 $4.70
Keywords: computational structures, common base language, programming languages
Abstract: This is the report on the work of the Computational Structures Group of Project MAC toward the design and specification of a common base language for programs and information structures. We envision that the meanings of programs expressed in practical source languages will be defined by rules of translation into the base language. The meanings of programs in the base language is fixed by rules of interpretation which constitute a transition system called the interpreter for the base language. We view the base language interpreter as the functional specification of a computer system in which emphasis is placed on programming generality — the ability of users to build complex programs by combining independently written program modules.

Our concept of a common base language is similar to the abstract programs of the Vienna definition method — but a single class of abstract programs applies to all source languages to be encompassed. The semantic constructs of the base language must be just those fundamental constructs necessary for the effective realization of the desired range of source languages. Thus we seek simplicity in the design of the interpreter at the expense of increased complexity of the translator from a source language to the base language. As an illustration of this philosophy, we present a rudimentary form of the base language in which nonlocal references are not permitted, and show how programs expressed in a simple block structured language may be translated into this base language.

The importance of representing concurrency within and among computations executed by the interpreter is discussed, and our approach toward incorporating concurrency of action in the base language is outlined. {AD 744-207}

TR-102
Hossley, R.
FINITE TREE AUTOMATA AND ω-AUTOMATA
Pages: 111 S.M. Thesis/September 1972 $6.65
Keywords: finite tree automata, automata
Abstract: Chapter I is a survey of finite automata as acceptors of finite labeled trees. Chapter II is a survey of finite automata as acceptors of infinite strings on a finite alphabet. Among the automata models considered in Chapter II are those used by McNaughton, Buchi, and Landweber. In Chapter II, we also consider several new automata models based on a notion of a run of a finite automaton on an infinite string suggested by Professor A. R. Meyer in private communication. We show that these new models are all equivalent to various previously formulated models.

M. O. Rabin has published two solutions of the emptiness problem for finite automata operating on infinite labeled trees. Appendices I and II contain a new solution of this emptiness problem. This new solution was obtained jointly by the author and Charles Rackoff. {AD 749-367}

TR-103
Sekino, A.
PERFORMANCE EVALUATION OF MULTIPROGRAMMED TIME-SHARED COMPUTER SYSTEMS
Pages: 280 Ph.D. Dissertation/September 1972 $11.75
Keywords: computer systems, performance, time-shared computers, operating systems, system response time, virtual memory, multiprogramming, multiprocessing, Multics
Abstract: This thesis presents a comprehensive set of hierarchically organized modular analytical models developed for the performance evaluation of multiprogrammed virtual-memory time-shared computer systems using demand paging. The hierarchy of models contains a user behavior model, a secondary memory model, a program behavior model, a processor model, and a total system model. This thesis is particularly concerned with the last three models. The program behavior model developed in this thesis allows us to estimate the frequency of paging expected on a given processing system. The processor model allows us to evaluate the throughput of a given multi-processor multi-memory processing system under multiprogramming. Finally, the total system model allows us to derive the response time distribution of an entire computer system under study.

Since all major factors (such as various system overhead times and idle times) which may decrease a system's computational capacity available for users' useful work are explicitly considered in the analyses using the above models, the performance predicted by these analyses is very realistic. A comparison of the performance of an actual system, the Multics system of M.I.T., and the corresponding performance predicted by these analyses confirms the accuracy of performance prediction by these models. Then, these analyses are applied to the optimization of computer systems and to the selection of the best performing system for a given budget. The framework of performance evaluation using these hierarchically organized analytical models guides human intuition in understanding the actual performance problems and provides us with reliable answers to most of the basic quantitative performance questions concerning throughput and response time of actual modern large-scale time-shared computer systems. {AD 749-949}

TR-104
Schroeder, M.D.
COOPERATION OF MUTUALLY SUSPICIOUS SUBSYSTEMS IN A COMPUTER UTILITY
Pages: 164 Ph.D. Dissertation/September 1972 $8.25
Keywords: computer utility, mutually suspicious subsystems, protection
Abstract: This thesis describes practical protection mechanisms that allow mutually suspicious subsystems to cooperate in a single computation and still be protected from one another. The mechanisms are based on the division of a computation into independent domains of access privilege, each of which may encapsulate a protected subsystem. The central component of the mechanisms is a hardware processor that automatically enforces the access constraints associated with a multidomain computation implemented as a single execution point in a segmented virtual memory. This processor allows a standard interprocedure call with arguments to change the domain of execution of the computation. Arguments are automatically communicated on cross-domain calls — even between domains that normally have no access capabilities in common. The processor, when supported by a suitable software system which is also discussed, provides the protection basis for a computer utility in which users may encapsulate data bases as protected subsystems, and then, without compromising the protection of the individual subsystems, combine protected subsystems of different users to perform various computations. {AD 750-173}

TR-105
Smith, B.J.
AN ANALYSIS OF SORTING NETWORKS
Pages: 93 Sc.D. Thesis/October 1972 $6.10
Keywords: sorting networks
Abstract: Comparators which sort two numbers can be interconnected to form networks which sort n numbers for any n. The input and output characteristics of comparator networks are analyzed from several different points of view. {AD 751-614}

TR-106
Rackoff, C.
THE EMPTINESS AND COMPLEMENTATION PROBLEMS FOR AUTOMATA ON INFINITE TREES
Pages: 44 S.M. Thesis/January 1973 $4.65
Keywords: automata, infinite tree automata, finite tree automata
Abstract: In [6] Rabin defines Automata on Infinite Trees, and the body of that paper is concerned with proving two theorems about these automata.

The result we consider in the first chapter says that there exists an effective procedure to determine, given an automaton on infinite trees, whether or not it accepts anything at all. We present a new decision procedure which is much simpler than Rabin's since we do not use an induction argument as he does. We show in Theorem 1, the main theorem of Chapter 1, that if ω is an automaton on infinite trees, then T(ω) (the set accepted by ω) is non-empty if and only if there exists a finite tree E and a run of ω on E of a particular type. This latter condition is equivalent to saying that the set accepted by a particular automaton on *finite* trees is non-empty. Hence (see Theorem 2) the emptiness problem for automata on finite trees can be reduced by Theorem 1 to the emptiness problem for automata on finite trees, which is shown decidable in [7]. Theorem 1 is proven by showing how maps on finite trees can generate maps on infinite trees which are then said to be finitely-generable. A corollary of the proof of Theorem 1 is that if an automaton on infinite trees accepts some input tree, then it accepts a finitely-generable one; this result was proved in a much more complicated way by Rabin in [5].

Chapter 2 is concerned with the more difficult result of [6] that for every automaton on infinite trees, ω, there exists another one, ω, such that ω' accepts precisely the compliment of the set accepted by ω. Rabin's construction of ω and the proof that it works is an involved induction. In this paper we present a fairly simple description of a complement machine ω', given ω, such that it is very plausible that ω' works in the sense that T(ω') = the compliment of T(ω). The proof that our construction works, however, is difficult and very similar in complexity to Rabin's proof in [6] that his (more difficult) construction works. {AD 756-248}

TR-107
Madnick, S.E.
STORAGE HIERARCHY SYSTEMS
Pages: 155 Ph.D Dissertation/April 1973 $8.00
Keywords: hierarchy, storage, storage hierarchy
Abstract: The relationship between page size, program behavior, and page fetch frequency in storage hierarchy systems is formalized and analyzed. It is proven that there exist cyclic program reference patterns that can cause page fetch frequency to increase significantly if the page size used is decreased (e.g., reduced by half). Furthermore, it is proven in Theorem 3 that the limit to this increase is a linear function of primary store size. Thus, for example, on a typical current-day paging system with a large primary store, the number of page fetches encountered during the execution of a program could increase 200-fold if the fpage size were reduced by half.

The concept of temporal locality versus spatial locality is postulated to explain the relationship between page size and program behavior in actual systems. This concept is used to develop a technique called the "tuple-coupling" approach. It is proven in Theorem 5 that storage system replacement algorithms, tuple-couple yields the benefits of smaller page sizes without the dangers of explosive page fetch activity.

Consistent with the results above and by generalizing conventional two-level storage systems, a design for a general multiple level storage hierarchy system is presented. Particular algorithms and implementation techniques to be used are discussed. {AD 760-001}

TR-108
Wand, M.
MATHEMATICAL FOUNDATIONS OF FORMAL LANGUAGE THEORY
Pages: 224 Ph.D. Dissertation/December 1973 $10.05
Keywords: language theory, finite tree automata
Abstract: We consider systems of functions closed under various operations. One such system, called a clone, provides an alternative to Lawvere's algebraic theories for the study of automata on finite trees. We introduce another such system, called a uclone, for the study of fixed-point operations incomplete lattices. We characterize free uclones in terms of regular trees in the lattice of flow diagrams. From this characterization three normal form theorems are derived. We then apply this framework to the study of formal languages by defining the notion of an equational class of languages. We show that most of the elementary properties of formal languages, such as normal form theorems, closure theorems, and pumping lemmas, may be derived from general theorems about the behavior of fixed points on lattices and do not depend upon combinatorial details. Similar results may be obtained for new classes of languages. For example, we obtain two quite different grammatical characterizations for a class of augmented context-free languages in which copying rules are allowed. This class is an equational class of languages which is not an AFL. We also prove that a variety of algebras is closed under relational extensions if and only if it is full linear presentable. Last we consider a class of systems, called k-models, in which restricted "Currying" rules are allowed. For these systems we prove a cut-elimination theorem. This theorem is proved by writing a computer program to convert derivation trees to the proper form, and then proving, using techniques developed for program-proving, that the program is total and correct.

TR-109
Johnson, D.
NEAR-OPTIMAL BIN PACKING ALGORITHMS
Pages: 401 Ph.D. Dissertation/June 1973 $15.35
Keywords: bin packing, algorithms, packing rule
Abstract: The Bin Packing problem is a model for a number of problems occurring in industry and computer science. Suppose we are given a list of pieces with sizes between 0 and 1, and a sequence of unit-capacity bins. Our goal is to pack the pieces into as few bins as possible. All known algorithms for finding optimal solutions to this problem require exponential time. In this thesis we study instead algorithms which generate near-optimal solutions and which run in low-order polynomial time.

The previously analyzed FIRST FIT and BEST FIT packing rules belong to a more general class of packing rules, the AAF packing rules, which we show all have the same worst case behavior. We extend these results to include the case when numbers in the input list are restricted to any given sub-interval of (0,1]. No algorithm which implements any packing rule in the class can use less than 0(nlogn) time, and we give implementations for several of the rules, including FIRST FIT and BEST FIT, which realize this bound. We then introduce linear time algorithms whose worst case behavior is the same as that of the above algorithms in many restricted situations and is never known to be worse.

It has previously been shown that if the input list is in decreasing order, FIRST FIT can asymptotically require no more than 5/4 times the optimal number of binds. We show that this result extends to an even larger class of algorithms, the AF algorithms, and generalize our proof to get results for lists with numbers in restricted ranges. We then show that in fact FIRST FIT and BEST FIT asymptotically require no more than 11/9 times the optimal number of bins, and this is the best bound possible. Other algorithms are suggested that may do even better.

Finally, we report the results of an empirical study using randomly generated lists to get a picture of the average case behavior of the algorithms studied. The average case behavior is very much better than our worst case results might have led one to expect. {PB 222-090}

TR-110
Moll, R.
COMPLEXITY CLASSES OF RECURSIVE FUNCTIONS
Pages: 95 Ph.D. Dissertation/June 1973 $6.20
Keywords: recursive functions, complexity
Abstract: An honest function is one whose size honestly reflects its computation time. In 1969 Meyer and McCreight proved the "honesty theorem", whic says that for every t, the t-computable functions are the same as the t'computable functions for some honest t'.

Ways of constructing honest functions are considered in detail. It is shown that for any t there is an honest t' such that the t-computable functions and the t'computable functions re the same, and such that t' is arbitrarily large on a dense set of arguments. Moreover any construction method satisfying certain natural criteria will (almost) have this property.

On the other hand it is shown that by relaxing these criteria we can guarantee that t' ≤ t on a (weak) dense set. We can also guarantee that t' will be bounded above by a predetermined recursive function on all but finitely many arguments. Finally, we show that in the case where t is monotone, t' can also be made monotone.

We consider the t-computable functions, and order these classes under set inclusion as t varies over the recursive functions. We show that given any total effective operator ***F*** and any recursive countable partial order R there is an r.e. sequence of machine running times T_0, T_1, ... T_n, ...such that if iRj, then the T_j computable functions properly contain the ***F***(T_i) computable functions, and if i and j are incomparable, then ***F***(T_i) >T_j infinitely often and ***F***(T_j) > T_i infinitely often. {AD 767-730}

TR-111
Linderman, J.P.
PRODUCTIVITY IN PARALLEL COMPUTATION SCHEMATA
Pages: 152 Ph.D. Dissertation/December 1973 $7.90
Keywords: program schemata, parallel computation, theory of programs
Abstract: A general model for parallel computation is developed in three parts. One part, the data flow graph, describes how actors which transform and test values are connected to the locations in a finite memory. Another part, an interpretation, supplies information about the contents of memory and the detailed nature of the transformations and tests.

The third part specifies how initiations and terminations of the actors are allowed to occur. We define this in a general way, using a set of sequences of initiation and termination events to model control. This allows us to prove results which apply to a broad class of control mechanisms.

Our major results are analogous to a theorem of Karp and Miller. Their theorem defines a class of schemata for which conflict-freeness is necessary and sufficient for determinacy. We use a weaker notion of determinacy which depends only upon the final contents of a subset of the memory locations. To establish necessity, we introduce the property of productivity which expresses whether individual transformations and tests contribute to the final results of a computation. {PB 226-159/AS}

TR-112
Hawryszkiewycz, I.T.
SEMANTICS OF DATA BASE SYSTEMS
Pages: 349 Ph.D. Dissertation/December 1973 $13.80
Keywords: data base systems, data base operations, concurrency
Abstract: A general data base system that allows tasks to be performed concurrently and users to share data bases is described. User operations on data bases are expressed in terms of an augmented relational data base model. Values of shared domains and relations can be changed by users but the changes permitted are such that data bases remain meaningful to all users.

Each user program defines a sequence of operations that cause transitions of the data base state. Operations for different users may be requested concurrently. Data base operations are defined in terms of their effect on the data base state. Outcomes of concurrent operations are defined to be correct if they are the same as if the operations had been performed in some merged sequence.

The definitions are then redefined in terms of an abstract model based on tree-structured objects, and closer to the level of practical implementation. A formal correspondence is developed between states of the data base system and states of the abstract model. Each operation is performed by a group of semantic procedures that transform the abstract state in a manner consistent with the transformation of the data base state.

The semantic procedures are designed so that concurrent execution of several data base operations is possible with a guarantee that the resulting abstract state is always consistent with some valid data base state. Thus an implementation that faithfully realizes the abstract model is proved to correctly realize that data base system. {PB 226-061/AS}

TR-113
Herrmann, P.P.
ON REDUCIBILITY AMONG COMBINATORIAL PROBLEMS
Pages: 32 S.M. Thesis/December 1973 $4.30
Keywords: combinatorial problems, reduction
Abstract: A large class of combinatorial problems have been shown by Cook and Karp to be computationally equivalent to within a polynomial. We exhibit some new problems in this class, and provide simpler proofs for some of the known reductions. {PB 226-157/AS}

TR-114
Metcalfe, R.M.
PACKET COMMUNICATION
Pages: 240 Ph.D. Dissertation/December 1973 $10.55
Keywords: packet communication, networks, computer networks
Abstract: This report develops a theory of packet communication; it analyzes users of computers in digital communication systems and examines structures for organizing computers in highly communicative environments. Various examples from existing computer networks, including the ARPA Computer Network and the ALOHA System, are used to motivate and substantiate analysis of (1) store-and-forward packet communication, (2) broadcast packet communication, and (3) distributed interprocess communication. {AD 771-430}

TR-115
Rotenberg, L.
MAKING COMPUTERS KEEP SECRETS
Pages: 393 Ph.D. Dissertation/February 1974 $15.10
Keywords: security, protection, authorization mechanisms, surveillance mechanisms, computer utility, proprietary services, computers and society, cryptography, cryptology
Abstract: This dissertation presents a unified design of protection mechanisms for a computer utility that (1) prevent accidental unauthorized releases of information, (2) prevent tyranny by dividing and limiting the power of the administrators of the utility, (3) preserve the independence of independent users of the utility, (4) accommodate to organizations having disparate traditional superior-subordinate relations, and (5) support proprietary services that allow users to build on the work of others in a context that protects the interests of lessors and lessees of services. The design includes specifications of both hardware and software protection mechanisms, including walls defined by domains and capabilities, and a hardware device, the Privacy Restriction Processor, that records the copying and combining of information in the computer by propagating privacy restrictions among restriction sets associated with segments and processes. The propagated restrictions prevent accidental unauthorized releases of information. But when a secret can be encoded into the timing or occurrence of system actions to prevent output of secrets, the encoded secret can escape. However, such escaping secrets can be detected and the offending computation can be arrested by the operating system.

The dissertation includes an analysis of the social concepts, systems, and conventions to which a computer utility is necessarily connected. The emergence of a 1984-like negative-utopia is shown to be a possible consequence of the on-going development of techniques for penetrating and taking over computer systems. {PB 229-352/AS}

TR-116
Stern, J.A.
BACKUP AND RECOVERY OF ON-LINE INFORMATION IN A COMPUTER UTILITY
Pages: 123 S.M. & E.E. Thesis/January 1974 $7.00
Keywords: backup, recovery, computer utility, Multics
Abstract: This thesis describes a design for an automatic backup mechanism to be incorporated in a computer utility for the protection of on-line information against accidental or malicious destruction. This protection is achieved by preserving on magnetic tape recent copies of all items of information known to the on-line file system. In the event of a system failure, file system damage is automatically assessed and missing information is recovered from backup storage. For isolated mishaps, users may directly request the retrieval of selected items of information. The design of the backup mechanism presented in this thesis is based upon an existing backup mechanism contained in the Multics system. As compared to the present Multics backup system, the new design lessens overhead, drastically reduces recovery time from system failures, eliminates the need to interrupt system operation for backup purposes, and scales up significantly better with on-line storage growth. {AD 774-141}

TR-117
Clark, D.D.
AN INPUT/OUTPUT ARCHITECTURE FOR VIRTUAL MEMORY COMPUTER SYSTEMS
Pages: 191 Ph.D. Dissertation/January 1974 $9.05
Keywords: input/output architecture, virtual memory, protection, architecture
Abstract: In many large systems today, input/output is not performed directly by the user, but is done interpretively by the system for him, which causes additional overhead and also restricts the user to whatever algorithms the system has implemented. Many causes contribute to this involvement of the system in user input/output, including the need to enforce protection requirements, the inability to provide adequate response to control signals from devices, and the difficulty of running devices in a virtual environment, especially a virtual memory. The goal of this thesis was the creation of an input/output system which allows the user the freedom of direct access to the device, and which allows the user to build input/output control programs in a simple and understandable manner. This thesis presents a design for an input/output subsystem architecture which, in the context of a segmented, paged, time-shared computer system, allows the user direct access to input/output devices. This thesis proposes a particular architecture, to be used as an example of a class of suitable designs, with the intention that this example serve as a tool in understanding the large number preferable form. {AD 774-738}

TR-118
Briabrin, V.
AN ABSTRACT MODEL OF A RESEARCH INSTITUTE: SIMPLE AUTOMATIC PROGRAMMING APPROACH
Pages: 22 March 1974 $4.00
Keywords: automatic programming, knowledge representation
Abstract: A problem of knowledge representation is considered in terms of designing a model for a simple sociological structure. A version of the access language ıs proposed which is based on three kind of expressions accepted by the system - constructors, specificators, and requests. In addition, some topics concerned with model implementation and extension are discussed. {PB 231-505/AS}

TR-119
Hammer, M. M.
A NEW GRAMMATICAL TRANSFORMATION INTO DETERMINISTIC TOP-DOWN FORM
Pages: 301 Ph.D. Dissertation/February 1974 $12.35
Keywords: parsing, grammatical transformation
Abstract: Although deterministic top-down parsing is an attractive parsing technique, the grammars to which it is applicable (the LL(k) grammars) are but a small subset of the LR(k) grammars, those that can be parsed deterministically bottom-up. In this thesis, the problem of transforming LR(k) grammars into equivalent LL(k) grammars is studied.

A new method of parsing, called multiple stack parsing, is introduced. This is a generalization of LR(k) parsing, containing a minimal infusion of top-down predictive techniques into deterministic bottom-up parsing. An automaton, the MSP(k) machine, which parses strings in this way is formally defined, and is shown to be equivalent to the canonical LR(k) parsing machine. A transformation procedure is described which constructs from M, a particular kind of MSP(k) machine for the grammar G (called a cycle-free machine), a new derived grammar $T_M(G)$. The grammar $T_M(G)$ generates the same language as the grammar G does and is (strong) LL(k) as well. No translating ability is lost in effecting this transformation, in the sense that $T_M(G)$ can support the same class of compilation activities, or syntax-directed translations, that G can.

The class of grammars which can be parsed by cycle-free MSP(k) machines and so are amenable to transformation, strictly includes both the LL(k) grammars, and the LC(k) (left corner parsable) grammars. Thus our transformation is more powerful than the previously available one of Rosenkrantz and Lewis. Furthermore, there are good algorithms applicable to many transformable grammars for constructing a cycle-free MSP(k) machine and making the entire transformation process more efficient; and the size of $T_M(G)$ can be systematically reduced without sacrificing its desirable qualities. {AD 775-545}

TR-120
Ramchandani, C.
ANALYSIS OF ASYNCHRONOUS CONCURRENT SYSTEMS BY TIMED PETRI NETS
Pages: 219 Ph.D. Dissertation/February 1974 $9.90
Keywords: Petri nets, timed Petri nets, pipelining, computation rate, performance analysis, distributed systems, concurrent systems
Abstract: This thesis is concerned with the modeling and performance analysis of systems which consist of concurrently acting components, an example of which is an asynchronous pipelined processor. The work is divided into two parts.

In the first part, a suitable model is developed for describing the structure of asynchronous concurrent systems. In conventional automata theory, the finite-state machine model is used to describe the behavior of systems; the problem with this is that a large number of states results when practical systems are modeled. In this thesis, each system component is modeled as a finite-state machine, and a system is viewed as an ensemble of interconnected finite-state machines. This has the advantage that the size of a system model grows linearly rather than exponentially with the number of system components. A subclass of Petri nets known as SMD (State Machine Decomposable) Petri nets is identified in order to formalize the notions of finite-state machines and their inter-connection. For convenience, systems of interest are divided into two broad categories:

(a) deterministic, or decision free,
(b) non-deterministic, or systems with decisions.

SMD Petri nets are used to model both classes of systems; in addition, a subclass of Petri nets known as LSP Petri nets is used to model those deterministic systems that cannot be modeled by SMD Petri nets.

The second part of the thesis is concerned with finding the computation rate of activities in real-world asynchronous concurrent systems. Practical systems are constructed from devices which have a finite speed of operation. Since Petri nets do not have time parameters as part of their definition, they can model the structure of systems but cannot be used to study their computation rate. The definition of Petri nets is augmented to model the speed of operation of a device in a system by assuming that the corresponding activity in the Petri net has a finite, non-zero time duration. The resulting nets are termed timed Petri nets, and methods are given for finding the computation rate of activities in timed SMD and LSP Petri nets. The results are applied to the analysis of several asynchronous systems drawn from areas within and outside the domain of computer systems. {AD 775-618}

TR-121
Yao, F.F.
ON LOWER BOUNDS FOR SELECTION PROBLEMS
Pages: 55 Ph.D. Dissertation/March 1974 $5.00
Keywords: selection problems
Abstract: Let $V_i(n)$ be the minimum number of binary comparisons that are required to determine the i-th largest of n elements drawn form a totally ordered set. In this thesis we use adversary strategies to prove lower bounds on $V_i(n)$. For $i = 3$, our lower bounds determine $V_3(n)$ precisely for infinitely many values of n, and determine $V_3(n)$ to within 2 for all n. For a general fixed i, our lower bound has the asymptotic form $n + (i-1)\log n - 0(\log(\log \cdot n))$ where $\log \cdot n$ is a very slowly growing function. As a result, the asymptotic behavior of $V_i(n)$ is determined to within $0(\log(\log \cdot n))$. A more general problem is raised in which one wants to find an element which is (i,j)-mediocre, i.e. smaller than at least i elements and greater than at least j elements. For $i = 1$, it is shown that the best algorithm is to select the $i + 1$st largest of any subset of $i + j + 1$ elements. It is an interesting question whether for general i this procedure is also optimal for finding an (i,J)-mediocre element. An affirmative answer to this question would imply $V_{n/2}(n) \leq 3n$. {PB 230-950/AS}

TR-122
Scherf, J.A.
COMPUTER AND DATA SECURITY: A COMPREHENSIVE ANNOTATED BIBLIOGRAPHY
Pages: 305 S.M. Thesis/January 1974 $12.50
Keywords: cryptography, data security, security
Abstract: Articles discussing computer and data security topics are scattered over a very large number of sources which publish articles on security on an irregular basis. This makes it quite difficult for the security consultant, the internal auditor, the computer user, the data processing manager, the business executive, or anyone else to find out what has actually been done in this field without doing extensive, time-consuming, literature research. To ease this problem there currently exist approximately seven computer security bibliographies containing from 50 to 250 entries. Although they are all less than three years old, only one has annotations over a few sentences in length, and only two use any sort of classification or index scheme. The one bibliography with paragraph length annotations is primarily concerned with very technical aspects of hardware and software access control. Most of the other bibliographies are also concerned with only certain subsets of security problems. This paper is apparently the first attempt to produce a bibliography covering all aspects of computer and data security, and having annotations that more than superficially describe each article's content.

This bibliography contains 1,022 entries. About half these entries are extensively annotated, another quarter being superficially annotated and the rest being unannotated. All extensively annotated entries are rated as to their current usefulness and uniqueness. A subject index of 160 items is provided for referencing purposes. The introduction to this bibliography briefly discusses: privacy, security, and integrity; threats of data misuse; development and scope of the bibliography; the subject index; outstanding articles and books, computer security firms; and the future. A list of 34 firms selling computer security services or equipment is presented following the bibliography. {AD 775-546}

TR-123
Saltzer, et al.
INTRODUCTION TO MULTICS
Page:203 February $9.40
Keywords: Multics, computer systems, computer utility, time-sharing
Abstract: The Multics project was begun in 1964 by the Computer Systems Research group of M.I.T. Project MAC. The goal was to create a prototype of a computer utility. This technical report represents the Introduction to the users manual for the Multics System. It is published in this form as a

convenient method of communications with researchers and students of computer system design. It is divided into three major parts: 1) Introduction to Multics, 2) Reference Guide to Multics and 3) Subsystems Writers' Guide to Multics. {AD 918-562}

TR-124
Laventhal, M.S.
VERIFICATION OF PROGRAMS OPERATING ON STRUCTURED DATA
Pages: 143 S.B. & S.M. Thesis/March 1974 $7.60
Keywords: verification, data structures
Abstract: The major method for verifying the correctness of computer programs is the inductive assertion approach. This approach has been limited in the past by the lack of techniques for handling data structures. In particular, ther has been a need for concepts with which to describe structured data during intermediate and final stages of a computation. This thesis describes an approach by which this problem can be handled, and demonstrates its use in proving several programs correct.

The key to the approach is the restriction of a data structure to a particular structural class. Primitive concepts are introduced which allow such a class to be concisely defined. Other concepts relate structures form a given class to data abstractions which the structures can be thought to represent. It is shown how to integrate the structural description with the actual proofs of correctness by incorporating results of general applicability into a logical formalism for a given structural class. {PB 231-365/AS}

TR-125
Mark, W.S.
A MODEL-DEBUGGING SYSTEM
Pages: 141 S.B. & S.M. Thesis/April 1974 $7.55
Keywords: debugging, automatic programming
Abstract: This research discusses a program which aids the user of an automatic programming system (APS) in the "debugging" of his model of his problem situation. In essence, the user must make sure that he and the APS mean the same thing by the description of the problem which the APS is to solve. The problem domain considered in this thesis is that of "business games" (i.e., the management simulation games which are used as a learning tool in the study of management). A language for describing models of these games is presented. The paper then describes the program's methods of simulating and finding bugs in models written in this language. Important aspects of the program's problem-solving approach to debugging are its internal knowledge of "bugs" and of user intention within the model. This internal knowledge stresses the importance of bugs arising from the interaction of submodels within the model. Some details of the program's implementation (in the Conniver language) are discussed. The necessity of "model debugging" in automatic programming is emphasized. {AD 778-688}

TR-126
Altman, V.E.
A LANGUAGE IMPLEMENTATION SYSTEM
Pages: 389 S.B. & S.M. Thesis/May 1974 $15.00
Keywords: language implementation, programming languages
Abstract: This paper presents the design and implementation of a particular Language Implementation System, (LIS), and investigates the utilization of that system in the development of artifical languages and their associated processors.

The Language Implementation System accepts the formal definition of the syntax (expressed in Backus Naur Form - BNF) and the semantics (expressed in the programming language PL/I) of an artificial language, and synthesizes a processor for that language. The parsers (lexical and primary) of the processor are highly efficient Deterministic Push Down Automata (DPDAs) computed form the language's CLR(k) gramMarch The CLR(k), (Comprehensive **L**eft to **R**ight, looking ahead **k** symbols) grammars are defined in the paper, and are shown to include virtually all "practical" artificial languages. The semantic interpreter of an artificial language is activated for a particular BNF rule whenever a syntactic construct defined by that rule is recognized during the parse of the language's input text.

Applications of the Language Implementation System are presented, and the system is shown to be applicable not only to "traditional" artificial languages such as PL/I, Algol, and Lisp, but also to interactive management information/decision system languages. Furthermore, the processors produced by LIS are not limited to traditional translators, but are also shown to be useful in developing complex Man-Machine decision systems in which they may be viewed as computational dispatchers or structured interfaces between the user and a more complex computational facility. {AD 780-672}

TR-127
Greenberg, B.
AN EXPERIMENTAL ANALYSIS OF PROGRAM REFERENCE PATTERNS IN THE MULTICS VIRTUAL MEMORY
Pages: 140 S.M. Thesis/May 1974 $7.55
Keywords: Multics, virtual memory, computer utility, paging
Abstract: This thesis reports the design, conducting, and results of an experiment intended to measure the paging rate of a virtual memory computer system as a function of paging memory size. This experiment, conducted on the Multics computer system at MIT, a large interactive computer utility serving an academic community, sought to predict paging rates for paging memory sizes larger than the existent memory at the time. A trace of all secondary memory references for two days was accumulated, and simulation techniques applicable to "stack" type page algorithms (of which the least-recently-used discipline used by Multics is one) were applied to it.

A technique for interfacing such an experiment to an operative computer utility in such a way that adequate data can be gathered reliably and without degrading system performance is described. Issues of dynamic page deletion and creation are dealt with, apparently for the first reported time. The successful performance of this experiment asserts the viability of performing this type of measurement on this type of system. The results of the experiment are given, which suggest models of demand paging behavior.

TR-128
Frankston, R.M.
THE COMPUTER UTILITY AS A MARKETPLACE FOR COMPUTER SERVICES
Pages: 100 S.M. & E.E. Thesis/May 1974 $6.35
Keywords: computer utility, computer services
Abstract: Computers are unique in their ability to be programmed for a wide variety of applications. This is in contrast with hardware dedicated to specific tasks such as the telephone system. Because of its flexibility, a computer system can support, concurrently, many diverse services that do not require dedicated hardware. Conversely, these services act to bring the capabilities of the computer to the consumer who might otherwise find the operational difficulty of running computer programs too formidable.

Since the computer is supporting many services which are sold to consumers it is natural to model the system as a marketplace for these services. Most contemporary computer systems are oriented towards users who run programs. The environment for services puts different requirements on the computer systems than do the needs of programmers, so as to permit all the participants in the market to make effective use of its facilities and without interfering with each other. As with any marketplace, it must be convenient to do business within its framework.

The requirements of such a marketplace are not satisfied in contemporary computer systems. However, the marketplace can be evolved from some existing computer systems without fundamental changes. Presently the use of a computer requires considerable expertise on the part of the user. The evolution to a marketplace is necessary if the capabilities of computer systems are to be made more widely available than they are now. {AD 780-436}

TR-129
Weissberg, R.
USING INTERACTIVE GRAPHICS IN SIMULATING THE HOSPITAL EMERGENCY ROOM
Pages: 192 May 1974 $9.10
Keywords: graphics, clinical decision making, medical applications
Abstract: The hospital emergency room is a complex system having many interrelated factors contributing to its operation. The emergency room administrator has limited control over certain of these factors: numbers of beds, nurses, doctors, x-ray units; for example. Other factors such as patient arrival rates and demands made upon available resources are largely uncontrollable. One of the main problems facing the emergency room manager is to find a reasonable balance among the many factors over which one has control in the face of a range of values of the factors over which little control is possible.

A computer program has been designed which uses computer graphics and interaction with the user to create a flexible modeling environment for analysis of hospital emergency rooms. In projects involving analysis of public systems, it is especially important that close communication be maintained between the public administrator and the analyst. Tools of the type which concern the present research can make a significant contribution towards this end.

The emergency room was chosen as the basis for the research for two reasons: First, the author had been a member of a team which performed an analysis of the Cambridge Hospital emergency room in Cambridge, Massachusetts, and therefore was somewhat familiar with the emergency room system and factors relevant to its analysis. Second, the emergency room is a system which in many hospitals is rapidly approaching a crisis: like the medical care system as a whole, the emergency room is experiencing profound changes in the demands being made of it. Patient arrival rates are increasing at an exponential rate. For many, the emergency room has become the primary source of medical care. Thus the very role of the emergency room is becoming unclear. The rapid changes in volume and nature of demand being experienced by the emergency room suggest that time invested in analysis and planning of the system would be well spent.

The program, the Tool for Interactive Graphical Emergency Room Simulation (which for ease of discussion is referred to as TIGERS) is a simulation-based modeling environment which has been implemented on the PDP-10 computer of the Programming Technology Division of Project MAC at M.I.T. This first effort, although general in scope, is based upon the emergency room at Cambridge Hospital. A preliminary model based upon this emergency room has been implemented. Valuable feedback has been obtained from Dr. Peter Mogielnicki there, and it is expected that other doctors in the Boston area may soon try out the system as well. The main thrust of the research is being concentrated not on designing a highly accurate model of a particular emergency room, but rather on development of a tool which can be used for such a purpose.

The actual implementation of the simulation within TIGERS involves the design of a model and the translation of the model into data bases and events. The task is made somewhat easier in that TIGERS provides all major data bases and several utility subroutines. The graphics updating is automatic, and routines are provided which make trivial the creation of light buttons for changing any relevant parameters.

The hardware upon which the present system is implemented is currently too expensive for practical application in most situations, but graphics technology is developing rapidly and is fast entering the realm of practicability for smaller installations. Both to the analyst and to the public administrator, the medium represents a potentially useful means of making simulation models more intuitive and easier to understand. {AD 780-437}

TR-130
Ruth, G.R.
ANALYSIS OF ALGORITHM IMPLEMENTATIONS
Pages: 271 Ph.D. dissertation/May 1974 $11.45
Keywords: algorithm implementations, program verification
Abstract: The thesis of this dissertation is that the intelligent analysis of algorithm implementations can be systematized and automated. In particular, it is shown how the correctness or near correctness of a program written to carry out a task according to known general plans can be systematically deduced. Emphasis is placed on understanding the workings of the program under analysis so that errors may be located, characterized and explained in programming terms.

Analysis consists of discovering the underlying plan of a program and interpreting it as a variation of the known algorithms for performing the program's task. The necessary knowledge of programming that serves as a basis for this is developed through an investigation of program synthesis. The means of translating intentions into programs — common computational and control mechanisms, procedure organization, and code generation — are studied. Analysis is then cast as the problem of finding the task realizing procedure that is equivalent, or most nearly so, to the actions of the observed program. The inclusion of common errors as intention implementation variations makes equivalence determination, and thus understanding, possible even in the case of faulty programs. As a check on the validity of these ideas a prototypical analysis program was written. It was applied to actual novice programs with satisfactory results.

The development of this analysis methodology illustrates several key factors in the procedure understanding process. Among these are the importance of good representations for the actions and structure of a procedure and for describing the state of the world at every step, the necessity of knowledge of the relationship between intentions and actions, the value of the synthesis-analysis dualism, and the utility of knowledge about common error types.

Possible applications of the principles and techniques presented here to computer assisted instruction, program verification, and general procedure analysis are discussed. {AD 780-408}

TR-131
Levin, M.
MATHEMATICAL LOGIC FOR COMPUTER SCIENTISTS
Pages: 177 June 1974 $8.65
Keywords: mathematics, logic, quantification theory, recursive functions
Abstract: This book is an introductory course in mathematical logic covering basic topics in quantification theory and recursive function theory, and is intended for the reader who is interested in artificial intelligence, computer linguistics, and other related areas. The text is theoretical, but organized with implementation in mind. Toward the end there are a few experimental subjects aiming toward systems that can examine their own behavior, and toward the semantics of programming languages. The arithmetization of metamathematics is carried out in LISP rather than in the natural numbers, following an axiomatic treatment of LISP.

TR-132
Janson, P.A.
REMOVING THE DYNAMIC LINKER FROM THE SECURITY KERNEL OF A COMPUTING UTILITY
Pages: 128 S.M. Thesis/June 1974 $7.15
Keywords: computer utility, security, protection, dynamic linker
Abstract: In order to enforce the security of the information stored in a computing utility, it is necessary to certify that the protection mechanism is correctly implemented so that there exist no uncontrolled access path to the stored information. Certification requires that the security kernel be much smaller and simpler than the supervisor of present general purpose operating systems. This thesis explores one aspect of improving the certifiability of a computing utility by designing a dynamic linker that runs outside the security kernel domain.

The dynamic linker is designed to run in any user protection domain of a multidomain computing utility. It is shown that the dynamic linker never needs the priveliges of the security kerel to properly operate. In particular, the thesis demonstrates the ability of the dynamic linker to link programs together across domain boundaries without violating the protection of either domain involved in the operation. {AD 781-305}

TR-133
Stockmeyer, L.J.
THE COMPLEXITY OF DECISION PROBLEMS IN AUTOMATA THEORY AND LOGIC
Pages: 224 Ph.D. Dissertation/July 1974 $10.05
Keywords: computational complexity, decision procedure, Turing machine
Abstract: The inherent computational complexity of a variety of decision problems in mathematical logic and the theory of automata is analyzed in terms of Turing machine time and space and in terms of the complexity of Boolean networks.

The problem of deciding whether a star-free expression (a variation of the regular expressions of Kleene used to describe languages accpeted by finite automata) defines the empty set is shown to require time and space exceeding any composition of functions exponential in the length of expressions. In particular, this decision problem is not elementary-recursive in the sense of KalMarch

The emptiness problem can be reduced efficiently to decision problems for truth or satisfiability of sentences in the first order monadic theory of (**N**,<), the first order theory of linear orders, and the first order theory of two successors and prefix, among others. It follows that the decision problems for these theories are also not elementary-recursive.

The number of Boolean operations and hence the size of logical circuits required to decide truth in several familiar logical theories of sentences only a few hundred characters long is shown to exceed the number of protons required to fill the known universe.

The methods of proof are analogous to the arithmetizations and reducibility arguments of recursive function theory. {PB 235-283/AS}

TR-134
Ellis, D.J.
SEMANTICS OF DATA STRUCTURES AND REFERENCES
Pages: 170 S.M. & E.E. Thesis/August 1974 $8.45
Keywords: data structures, semantic models for programming languages, type checking, programming language semantics
Abstract: Each programming language that handles data structures has its own set of rules for working with them. Notions such as assignment and construction of structures values appear in a huge number of different and complicated versions. This thesis presents a methodology which provides a common basis for describing ways in which programming languages deal with data structures and references to them. Specific concern is paid to issues of sharing.

The methodology presented here consists of two parts. The base language model, a formal semantic model introduced by Dennis, is used to give the work here a precise foundation. A series of "mini-languages" are defined to make it simpler for a variety of constructs found in contemporary programming languages. {PB 236-594/AS}

TR-135
Pfister, G.F.
THE COMPUTER CONTROL OF CHANGING PICTURES
Pages: 267 Ph.D. Dissertation/September 1974 $11.35
Keywords: picture modules, daemons, changing pictures
Abstract: This document describes DALI (Display Algorithm Language Interpreter), a special-purpose programming language for the creation and control of changing pictures which exhibit complex static and dynamic interactions among their elements. DALI allows complex organizations of interpolated ("smooth") change, discrete change, and change in the structure of a picture to be generated in a modular way, in the sense that picture elements determine their own behavior and hence manner of change.

In DALI, pictures are composed of elements called *picture modules*. These are analogous to procedural activations of processes, and contain arbitrary event driven procedures called "daemons". Daemons are run under the control of global scheduling rules based on the functional dependence of daemons on one another. These rules result in smooth inter-daemon (process) communication and cooperation with no implicit or explicit reference to semaphores or other synchronization primitives in user code, while at the same time providing for a high degree of parallelism. Circular inter-daemon functional dependence results in iteration or relaxation. The environment structure used is predominantly stack-oriented. {AD 787-795}

TR-136
Ward, S.
FUNCTIONAL DOMAINS OF APPLICATIVE LANGUAGES
Pages: 121 Ph.D. Dissertation/September 1974 $6.95
Keywords: applicative languages, lambda calculus
Abstract: The expressive power of a particular applicative language may be characterized by the set of abstract functions directly representable in that language the common FUNARG and applicative order problems are scrutinized in this way, and the effects of these weaknesses are related to the inexpressibility of classes of functions.

Certain computable functions which are inexpressible in the lambda calculus are identified, and it is established that the interpretations of these functions requires a mechanism fundamentally equivalent to multiprocessing. the EITHER construct is proposed as an extension to the lambda calculus, and several theories including this mechanism are presented and proved consistent (in the sense that they introduce no new equivalences into the lambda calculus).

A syntactic analog to the Scott construction, *-conversion, is developed in conjunction with these theories; this adjunct allows reduction of expressions having no normal forms in the usual lambda calculus to finite normal form approximations of the expressions. This leads naturally to a technique for proving the extensional equivalence of lambda calculus expressions which are not interconvertible. {AD 787-796}

TR-137
Seiferas, J.
NONDETERMINISTIC TIME AND SPACE COMPLEXITY CLASSES
Pages: 121 Ph.D. Dissertation/September 1974 $6.95
Keywords: computational complexity, Turing machine, theory of computation
Abstract: The marginal utility of the Turing machine computational resources running time and storage space are studied. A technique is developed which, unlike diagonalization, applies equally well to nondeterministic and deterministic automata. For f, g time or space bounding functions with $f(n+1)$ small compared to $g(n)$, it is shown that, in terms of word length n, there are languages which are accepted by Turing machines operating within time or space $g(n)$ but which are accepted by no Turing machine operating within time or space $f(n)$. The proof involves use of the recursion theorem together with "padding" or "translational" techniques of formal language theory.

Relations between worktape alphabet size, number of worktape heads, number of input heads, and Turing machine storage space are established. Within every common subexponential space bound, it is shown that enlarging the worktape alphabet always increases computing power. A hierarchy of two-way multihead finite automata is obtained even in the nondeterministic case.

Results that are only slightly weaker are obtained for Turing machines that accept only languages over a one-letter alphabet. {PB 236-777.AS}

TR-138
Yun, David Y. Y.
THE HENSEL LEMMA IN ALGEBRAIC MANIPULATION
Pages: 262 Ph.D. Dissertation/November 1974 $11.20
Keywords: algebraic manipulation, factorization, polynomial operation
Abstract: New and improved algorithms for computation in several fundamental polynomial operations are presented. The common bases for these algorithms are generalizations of the p-adic technique used in the constructive proof of the Hensel Lemma. Multivariate polynomial operations are stressed due to the special importance of the multivariate Hensel-type construction in replacing the modular evaluation-and-interpolation technique under certain conditions. Due to the availability of numerous (not completely satisfactory) methods for the computation of polynomial greatest common divisors (GCD) the EZGCD algorithm based on the Hensel construction is given special emphasis. An intuitive computing time analysis and many empirical experiments are made to compare the performance of this algorithm with two other major methods, especially the Modular GCD Algorithm. Both theoretically and by actual computing data, the new EZGCD Algorithm demonstrates promising efficiencies by taking advantage of the sparseness of multivariate polynomials. An intuitive and more 'engineering" approach to computing time analysis also appears to give quite accurate predictions of actual run times for many practical problems. Other applications of the Hensel-type constructions, resulting in improved algorithms for computing polynomial factorizations, contents and primitive parts, and square-free decompositions are also described. {AD A002-737}

TR-139
Ferrante, J.
SOME UPPER AND LOWER BOUNDS ON DECISION PROCEDURES IN LOGIC
Pages: 268 Ph.D. Dissertation/November 1974 $11.35
Keywords: logic, computational complexity, theory of computation, Turing machine
Abstract: The computational complexity of some decidable formal theories in logic is classified in terms of the amount of time or space needed to decide the theory.

For the theories of one successor, order on the nonnegative integers, well-order and lexicographical order, we obtain a nondeterministic lower bound of linear space and a deterministic upper bound on space of order n^3, for the theory of rational order, we establish a deterministic upper bound on space of order $n \log_2 (n)$. This is close to the known lower bound of nondeterministic space $\sqrt{n}$.

For the theory of a 1-1 unary function and the theory of two successors, we establish a lower bound of nondeterministic exponential time and an upper bound of deterministic exponential space. Finally, we obtain a nondeterministic, doubly exponential lower bound on space for the theory of a 1-1 unary function with a monadic predicate and the theory of one successor with a monadic predicate.

The upper bounds on complexity are obtained by the technique of Ehrehfeucht games; the lower bounds by efficient arithmetization of time-space-bound Turing machines. {PB 238-121/AS}

TR-140
Redell, D.D.
NAMING AND PROTECTION IN EXTENDABLE OPERATING SYSTEMS
Pages: 161 Ph.D. Dissertation/November 1974 $8.15
Keywords: protection, operating systems, extendable operating systems, mutually suspicious subsystems
Abstract: The properties of capability-based extendable operating systems are described, and various aspects of such systems are duscussed, with emphasis on the conflict between free distribution of access privileges and later revocation of those privileges. The discussion culminates in a set of goals for a new capability scheme.

A new design is then proposed, which provides both type extension and revocation through the definition of generalized sealing of capabilities. The implementation of this design is duscussed in sufficient detail to demonstrate that it would be workable and acceptable economical.

The utility of the proposed capability mechanism is demonstrated by describing two facilities implementable in terms of it. These are: (a) revocable parameters for calls between mutually suspicious subsystems, and (b) directories providing a civilized medium for the storage and distribution of revocable capabilities. {AD A001-721}

TR-141
Richards, M., Evans, A. Jr., Mabee, R.F.
THE BCPL REFERENCE MANUAL
Pages: 53 December 1974 $4.90
Keywords: BCPL, programming languages, compiler writing
Abstract: BCPL is a language which is readable and easy to learn, as well as admitting of an efficient compiler capable of generating efficient code. It is made self consistent and easy to define accurately by an underlying structure based on a simple idealized object machine. The treatment of data types is unusual and it allows the power and convenience of a language with dynamically varying types and yet the efficiency of FORTRAN. BCPL has been used successfully to implement a number of languages and has proved to be a useful tool for compiler writing. The BCPL compiler itself is written in BCPL and has been designed to be easy to transfer to other machines; it has already been transferred to more than ten different systems. {AD A003-599}

TR-142
Brown, G.P.
SOME PROBLEMS IN GERMAN TO ENGLISH MACHINE TRANSLATION
Pages: 189 S.M. & E.E. Thesis/December 1974 $9.00
Keywords: language translation, translation, natural language
Abstract: This paper discusses some problems in the machine translation of natural language, in particular, for translation from German into English. An implementation of some parts of the translating process has been built. The system consists of a German interpretive grammar, to take in German text and output a set of semantic representations, and a generator, to produce English sentences from single semantic representations. Although based on the assumption that understanding is necessary for correct translation of text, the system does not now contain an understanding componnt to choose between semantic representations. The representation of knowledge and its use in natural language understanding is a research area that is already under intensive investigation elsewhere. The implementation described here is based on a systemic grammar analysis of German and English, and it applies and extends the work of Winograd. Special attention is paid to questions of semantic representation in a multi-language setting and to stylistic issues in English generation. {AD A003-002}

TR-143
Silverman, H.
A DIGITALIS THERAPY ADVISOR
Pages: 98 S.M.Thesis/January 1975 $6.25
Keywords: clinical decision making, medical applications, digitalis
Abstract: The physician administering digitalis makes use of the full richness of the clinical setting to form his/her impressions and decide on a therapeutic program. The weakness of existing programs which formulate digitalis dosage regimens lies in their inability to use all of the clinical data available - both quantitative and qualitative. This report describes the construction of a computer system which formulates digitalis dosage regimens and which adjusts this regimen by interpreting the patients response to the original dosage regimen.

TR-144
Rackoff, C.
THE COMPUTATIONAL COMPLEXITY OF SOME LOGICAL THEORIES
Pages: 130 Ph.D. Dissertation/February 1975 $7.25
Keywords: computational complexity, logic, theory of computation
Abstract: Upper and lower bounds on the inherent computational complexity of the decision problem for a number of logical theories are established.

A general form of Ehrenfeucht game technique for deciding theories is developed which involves analyzing the expressive power of formulas with a given quantifier depth. The method allows one to decide the truth of sentences by limiting quantifiers to range over finite sets. In particular for *the theory of integer addition* an upper bound of space $2^{2^{cn}}$ is obtained; this is close to the known lower bound of nondeterministic time $2^{2^{c'n}}$.

A general development of decision procedures for theories of product structures is presented, which allows one to conclude in most cases that if the theory of a structure is elementary recursive, then the theory of its weak direct power (as well as other kinds of direct products) is elementary recursive. In particular, for *the theory of the weak direct power of* $<N, +>$, and hence for integer multiplication, an upper bound of space $2^{2^{2^{cn}}}$ is obtained. The known lower bound is nondeterministic time $2^{2^{2^{c'n}}}$

Finally, the complexity of the theories of pairing functions is discussed and it is shown that no collection of pairing functions has an elementary recursive theory.

TR-145
Henderson, D.A.
THE BINDING MODEL: A SEMANTIC BASE FOR MODULAR PROGRAMMING SYSTEMS
Pages: 282 Ph.D. Dissertation/February 1975 $11.80
Keywords: programming system, modularity
Abstract: A *programming system* is a computer system which supports a community of programmers who can make use of one another's work. A *module* is a (possibly complex) construction usually comprising both programs and data which will provide some service. A programming system is said to be *modular* if modules can be constructed within the system from existing modules. In such a system: mechanisms must exist which permit any module to be used by any other (the system must be *flexible*); modules must be responsible for supplying all the modules they need in order to realize their behaviors (modules must be *self-sufficient*); modules must not conflict with one another when used together (sets of modules must be *compatible*); and module behavior must be independent of the identity of the modules which invoke the modules (modules must be *non-discriminatory*).

For various reasons, existing programming systems are not modular.

The Binding Model is an abstract machine designed as an "ideal" kernel for modular programming systems. It is based on the notions of "location-less" data, unrestricted structuring, limited mutability, explicit binding, and controlled access to data.

The value of the model is established in two ways: Firstly, examples are used to demonstrate that common constructs of programming languages and operating systems are realizable in the model. Secondly, it is shown that two kinds of desirable special behavior can be given formal definitions in the model, and that there are large, structurally-defined classes of programs which satisfy these definitions. {AD A006-961}

TR-146
Malhotra, A.
DESIGN CRITERIA FOR A KNOWLEDGE-BASED ENGLISH LANGUAGE SYSTEM FOR MANAGEMENT: AN EXPERIMENTAL ANALYSIS
Pages: 355 Ph.D. Dissertation/February 1975 $14.00
Keywords: natural language, knowledge based systems, expert systems, parsing, management support system
Abstract: This thesis investigates the utility and feasibility of a knowledge-based English language computer system to support management. An "ideal" system was designed to contain knowledge about a problem-domain and respond to questions and commands phrased in natural English. A prototype was implemented based upon the corporate data base of a hypothetical manufacturer of lead batteries.
To investigate actual system usage a "perfect" English language system was simulated with the assistance of the prototype. This was capable of responding to requests in free English typed in at a computer terminal. Twenty three subjects were asked to solve a problem involving the battery manufacturer using this system.

The experimenter showed that managers were able to start quickly and work naturally with a system that could respond to requests phrased in English and could provide information about itself. Analysis of the words used in the sentences seems to indicate that a vocabulary of 1000 to 1500 words may be adequate for a domain-specific system. Some 78$SM of the sentences used by the managers fell into ten basic syntactic types and a moderately powerful parser would seem to be able to provide an adequate capability. To reach some understanding of the amount of knowledge required in a domain-specific system the subjects' requests were also analyzed for the knowledge that would be required to respond to them. We found thats although the amount of knowledge required is large, it is feasible to incorporate it in a management-support system.

The problem-solving protocols obtained through the experiment were used to test a "frame oriented" paradigm of problem-solving which states that managers analyze problems by checking hierarchical lists of potentially contributing subproblems. The data supports the paradigm with some evidence of exceptional behavior. This strengthens the generality of our results.

The final section of the thesis presents a design for an English language management-support system that is both technologically feasible and managerially useful.

TR-147
Van De Vanter, M.L.
A FORMALIZATION AND CORRECTNESS PROOF OF THE CGOL LANGUAGE SYSTEM
Pages: 99 S.M. Thesis/March 1975 $6.30
Keywords: programming languages, CGOL
Abstract: In many important ways the design and implementation of programming languages are hindered rather than helped by BNF. We present an alternative meta-language based on the work of Pratt which retains much of the effective power of BNF but is more convenient for designer, implementor and user alike. Its amenability to formal treatment is demonstrated by a rigorous correctness proof of a simple implementation.

TR-148
Johnson, J.
PROGRAM RESTRUCTURING FOR VIRTUAL MEMORY SYSTEMS
Page: 282 Ph.D. Dissertation/March 1975 $11.80
Keywords: virtual memory, program restructuring, paging
Abstract: The problem area addressed in this report is program restructuring, a method of reordering the relocatable sectors of a program in its address space to increase the locality of the programs reference behavior, thereby reducing rhe number of page fetches required for its execution in a virtual memory system.

Theoretical upper and lower (optimum) bounds are derived for the paging performance of programs over all partitions of relocatable sectors into pages.

Program restructuring techniques are developed which use intersector reference models based on sector working sets and sector stack distances. These intersectors reference models identify the local reference behavior, and clustering procedures are developed that use this local reference behavior to rearrange sectors into pages such that significant improvement in paging performance is obtained.

Results of measurements of paging performance obtained in the computer laboratory are discussed. The relationship between the paging performance of a program restructured by the practical restructuring algorithms and the theoretical bounds on paging performance are compared. {AD A009-218}

TR-149
Snyder, A.
A PORTABLE COMPILER FOR THE LANGUAGE C
Pages: 74 S.B. & S.M. Thesis/May 1975 $5.55
Keywords: compilers, C programming language, portable compilers, code generation, machine descriptions, abstract machines, implementation languages
Abstract: This paper describes the implementation of a compiler for the language **C**. The compiler has been designed to be able to be capable of producing assembly-language code for most register-oriented machines with only minor recoding. Most of the machine-dependent information used in code generation is contained in a set of tables which are constructed automatically from a machine description provided by the implementor. In the machine description, the implementor models the target machine by defining a machine-dependent abstract machine for which the code generator produces intermediate code. The abstract machine is abstract in that it is a **C** machine: its registers and memory are defined in terms of primitive **C** data types and its instructions perform basic **C** operations. The abstract machine is machine-dependent in that there is a close correspondence between the registers of the abstract machine and those of the target machine, and between the behavior of the abstract machine instructions and the corresponding target machine instructions or instruction sequences. The implementor defines the corresponding target machine instructions or instruction sequences. The implementor defines the corresponding target machine instructions or instruction sequences. The implementor defines the translation from an abstract machine program to a target machine program by providing in the machine description a set of simple macro definitions for the abstract machine instructions. In addition, macro definitions may be provided in the form of **C** routines where additional processing capability is needed. {AD A010-218}

TR-150
Rumbaugh, J.E.
A PARALLEL ASYNCHRONOUS COMPUTER ARCHITECTURE FOR DATA FLOW PROGRAMS
Pages: 319 Ph.D. Dissertation/May 1975 $12.90
Keywords: data flow programs, architecture, parallel computation, data flow languages, data flow machine
Abstract: This thesis involves the design of a parallel programming language and a parallel processor computer that runs programs expressed in that language. Sequencing of instruction execution in the Data Flow Language depends only on the availability of operands required by instructions. Data flow instructions have no side-effects; therefore unrelated instructions can be executed concurrently without interference if each has its required operands. In spite of the presence of concurrency, programs that meet a specified syntactic rule are guaranteed to have determinate behavior. Data flow procedures define functions from argument values to result values.

The Data Flow Machine is hierarchically constructed as a network of simple modules. All module interactions are asynchronous. The principal working elements of the machine are a set of activation processors, each of which performs the execution of one activation of data flow procedure held in a local memory within the processor. Because data flow operations have no side-effects, each processor can operate independently of the others. A pipeline of logical units within each processor performs the execution of several concurrently active instructions. All operations of a data flow procedure are performed by a processor except procedure calls, which cause the creation of new activations in other processors, and operations on structured values, which are performed by structure controller modules using values stored in a central Structure Memory. Concurrency within a data flow procedure provides something to do while a procedure call or structure operation is outstanding.

The behavior of the machine is specified by a formal description language. A proof that the machine correctly implements the language is presented.

The principal advantages of the Data Flow Machine over conventional multiprocessors are reduced complexity of the processor-memory connection, simple representation and implementation of concurrent activity, and guaranteed determinacy of programs. {AD A010-918}

TR-151
Manning, F.
AUTOMATIC TEST, CONFIGURATION AND REPAIR OF CELLULAR ARRAYS
Pages: 243 Ph.D. Dissertation/June 1975 $10.60
Keywords: cellular arrays, programmable logic, VLSI, computer-maintained machines, fault-tolerant digital machines
Abstract: A cellular array is an itterative array of identical information processing machines, cells. The arrays discussed are rectangular arrays of programmable logic, in which information stored in a working cell tells the cell how to behave. No signal line connects more than a few cells. A loading mechanism is each cell allows a computer directly connected to one cell to load any good cell that is not walled off by flawed cells. A loading arm is grown by programming cells to form a path that carries loading information. Cell mechanisms allow a computer to monitor the growth of a loading arm, and to change the arm's route to avoid faulty cells. Properly programmed cells carry test signals between a twisted cell and a testing computer directly connected to only a few cells. The computer may discover the faulty cells in an array; and repair the array by loading the array's good cells to embed a desired machine.
Terminology and network models are developed to describe the characteristics of a machine that are important to the test and repair of an array embedding in that machine. Important machine classes are defined, and their test and repair requirements are compared. Computer simulations of repair aid this comparison.

Each machine class is represented by a particular cellular machine design. Arrays are presented for realizing highly-integrated, computer-maintained memories, such as variable-length shift registers, random-access memories, and track-addressed sequential-access memories. One flawed array of simple cells may perform like any digital machine, within limits set by the size of the array, its number of input-output leads, and the speed of its components. One such machine can test, configure, and repair its cellular environment. Applications for these cellular arrays are discussed.

The thesis' approach is oriented toward the realities and trends in large-scale integrated circuit production; and has potential integration level, reliability, maintainability, and flexible advantages. {AD A012-822}

TR-152
Qualitz, J.E.
EQUIVALENCE PROBLEMS FOR MONADIC SCHEMAS
Pages: 149 Ph.D. Dissertation/June 1975 $7.80
Keywords: equivalence, monadic schemas

Abstract: A class of monadic program schemas is defined. This class, called iteration schemas, consists of schemas whose programs comprise assignment statements, conditional statements, and iteration statements. These schemas are shown to correspond to program schemas which are structured, and are shown to be strictly less "powerful" than the monadic program schemas.

A notion of equivalence is formalized as the functional equivalence of schemas under free interpretations, interpretations which represent symbolically the set of all interpretations of a schema. It is shown that the equivalence problem for iteration schemas is unsolvable, even if the schemas posses highly restrictive properties. questions are raised regarding the decidability of equivalence for various subclasses of iteration schemas, and equivalence is shown to be decidable for several of these classes.

The equivalence problems for structured independent location schemas are examined in particular detail. A weak form of equivalence is shown to be undecidable for the schemas, and the general equivalence problem is shown to be related in a non-trivial manner to the equivalence problem for multi-tape finite automata. {AD A012-823}

TR-153
Miller, P.B.
STRATEGY SELECTION IN MEDICAL DIAGNOSIS
Pages: 130 S.M. Thesis/September 1975 $7.25
Keywords: clinical decision making, expert systems, medical applications, strategic selection
Abstract: The recorded, verbal problem-solving behavior of doctors performing the diagnostic task of taking a present illness was analyzed in this research. The goal of the analysis was to discover what data-acquisition strategies were used by the doctors to accomplish the task. A model called the strategy frame model was created to describe the strategies that were found and to provide a mechanism for the selection of a strategy. In this model strategy selection is determined by the problem space of the doctor - his internal diagnostic configuration. A scheme for classifying strategies as confirmation, elimination, description or exploration was also developed.

TR-154
Grief, I.
SEMANTICS OF COMMUNICATING PARALLEL PROCESSES
Pages: 161 Ph.D. Dissertation/September 1975 $8.15
Keywords: semantics, parallel processing
Abstract: The thesis of this dissertation is that an understanding of the ordering constraints that are introduced among events of parallel process is essential to the understanding of synchronization and that therefore any language for specifying synchronization of parallel process should be based on a theory of such orderings. While it is possible to write specifications for systems of communicating parallel processes by reference to the time ordering of some global clock external to the system, such specifications cannot be as useful as ones which are in terms of orderings derivable within the system. Specifications should place constraints on intended behavior of the computer system itself rather than on the possible observations of the system's behaviors from some global viewpoint which may in fact be totally unrealizable.

The dissertation is a development of a specification language. It is based on a model of computation in which an individual process is represented by a totally ordered set of events. Synchronization properties of systems of independent processes are guarantees that in fact the set of events in the system can be ordered by a partial order which properly contains the union of the processes' total orders. This system ordering can be caused by the presence in a system of side-effect primitives or of synchronization primitives. Thus this model applies equally well both to busy waiting synchronization based on coordinated use of storage cells by independent processes and to non-busy waiting synchronization such as that induced by semaphores and structured synchronization primitives. In addition to applying to a range of types of synchronization, the specification language is also used to define a programming language. The meaning of a program is the specification of the behavior of the system into which that program is compiled. Specifications can be written for synchronization problems and for their implementations in terms of various primitives. {AD A016-302}

TR-155
Kahn, K.M.
MECHANIZATION OF TEMPORAL KNOWLEDGE
Pages: 153 S.M. Thesis/September 1975 $7.90
Keywords: temporal knowledge, time, medical diagnosis
Abstract: The design and implementation of a collection of computer programs knowledgable about time "in general", called the time specialist, is described. The thesis that this time specialist can be placed in the service of larger more general problem solvers is demonstrated for two examples, medical diagnosis and the understanding of a time-travel story.

The time specialist accepts a wide variety of facts and questions relating to the time of events. These include dates, vague terms such as "a few weeks ago", and two kinds of intervals. The "fuzziness" or inexactness of the time of events is handled differently for each of the representation types. The time specialist contains routines that compare, combine and translate between these various representation types.

The time specialist attempts to maintain a consistent data base. As facts are entered into the system, they are checked for their consistency with previously accepted facts. The time specialist corrects, to the extent possible, the data base after previously believed facts are doubted.

Incoming facts are organized by the time specialist to facilitate inference. Events are organized by their dates, by their position in a sequence of events, and by their relation to other very common events such as "now" and "birth". Routines that create, maintain, correct and use these organizational structures are described. The importance of organizing principles for facts is indicated.

TR-156
Bratt, R.G.
MINIMIZING THE NAMING FACILITIES REQUIRING PROTECTION IN A COMPUTING UTILITY
Pages: 129 S.M. Thesis/September 1975 $7.20
Keywords: computer utility, security kernel, Multics
Abstract: This thesis examines the various mechanisms for naming the information objects stored in a general-purpose computing utility, and isolates a basic set of naming facilities that must be protected to assure complete control over user interaction and that allow desired interactions among users to occur in a natural way. Minimizing the protected naming facilities consistent with the functional objective of controlled, but natural, user interaction contributes to defining a security kernel for a general-purpose computing utility. The security kernel is that complex of programs that must be correct if control on user interaction is to be assured.

The Multics system is used as a test case, and its segment naming mechanisms are redesigned to reduce the part that must be protected as part of the supervisor. To show that this smaller protected naming facility can still support the complete functionality of Multics, a test implementation of the design is performed. The new design is shown to have a significant impact on the size and complexity of the Multics supervisor.

TR-157
Meldman, J.A.
A PRELIMINARY STUDY IN COMPUTER-AIDED LEGAL ANALYSIS
Pages: 215 Ph.D. Dissertation/November 1975 $9.80
Keywords: computer-aided legal analysis, jurismetrics, expert systems
Abstract: This paper describes the prototype for a computer system that can perform a simple kind of legal analysis. The system user, who is presumed to be a lawyer, describes to the system a hypothetical set of facts. The system determines the extent to which these facts fall within certain legal doctrines (by syllogism), or near to these doctrines (by analogy). During this process, the system may ask the user for additional facts. The system then tells the user of its determinations and of the logic behind its conclusions with reference to judicial decisions and other legal authority. The prototype system communicates with the user in a computer language (called Preliminary Study Language) designed to be translatable into and out of English but natural-language processing techniques, based on case grammar, that are currently being developed in other research.

As the basis for this analysis, strucutal machine models are built to represent legally-relevant human activity and doctrines of law. The primitive components in these structures represent simple things and relations (like persons, firearms, hitting, near, etc.) in the everyday world of human affairs. These things and relations are classified hierarchically into categories. They are assempled into facts comprising two things and the relation between them. Facts, in turn, are assembled into more complicated structures called situations, which are represented in terms of component elements, or in terms of alternative types, or both. These situational structures are used to represent the hypothetical facts being analyzed as well as the factual content of legal doctrines. The factual situations of specific cases provide examples and counter-examples that behave as alternative types of the situational components of more general legal doctrine. The prototype system contains representations for doctrine involving civil battery and assault.

Analysis is performed by decomposing the situations that represent legal doctrines according to their elements and their types. When this decomposition reaches the level of things and relations, these things and relations, together with their situational structure, are matched against the things and relations contained in the hypothetical facts. The matching of individual things and relations is accomplished by reference to their hierarchical categorization. {AD A018-997}

TR-158
Grossman, R.W.
SOME DATA BASE APPLICATIONS OF CONSTRAINT EXPRESSIONS
Pages: 156 S.M. Thesis/February 1976 $8.00
Keywords: data base, constraint expressions, knowledge representation
Abstract: This report presents a novel network-like representation for information, called "constraint expressions" (CE). CE makes use of some of the knowledge-representation techniques developed by Artificial Intelligence research. A CE network consists of points (which represent classes of objects) interconnected by constraints (which represent the relationships which are known to hold among the classes). All constraints are defined in terms of six primitive ones. The data in a CE network is accessed by propagating various kinds of labels through it: Each constraint can be viewed as an active process which looks for certain patterns of labels on some of its attached points, and then propagates new labels to other points when such patterns occur.

The CE representation provides several significant features which are not found in most current data models. First, the same mechanism is used to represent "general" as well as "specific' information. For example, "The sex of Jane Smith is female" is specific, while "Every person has a unique sex which is either 'male' or 'female'" is general.

Second, CE's label-propagation procedure implements logical consistency checking: Data base integrity can be maintained by checking all new data for consistency with the existing information. Since the data base can contain general information (representing a "semantic model" of the data base's application domain), new specific data can be rejected if it is inconsistent with either other specific data or with the general information. Also, the general information can itself be checked for internal consistency.

Third, the CE representation is sufficiently modular and well-defined so that is has a precise formal semantics, which insures that CE's definition contains no hidden ambiguities or contradictions.

Fourth, CE's modularity allows the label propagations to be done in parallel, so that parallel hardware can be used to full advantage. {AD A024-149}

TR-159
Hack, M.
PETRI NET LANGUAGES
Pages: 128 March 1976 $7.15
Keywords: Petri nets, programming languages, automata
Abstract: In a labeled Petri Net we assign symbols from an alphabet to some or all the transitions of a Petri Net. To each firing sequence of such a Labeled Petri Net corresponds a string over the alphabet. We study the languages obtained in this way by all firing sequences of a Petri Net, or by all firing sequences which reach a given final marking. We consider the closure properties of these languages, their characterization, their relation to other language families, and the decidability of various problems concerning these languages. The last chapter relates Petri Nets to Counter Automata and Weak Counter Automata, introduces Inhibitor Nets and Priority Nets, and considers extensions and limitations of the Petri Net Languages.

TR-160
Bosyj, M.
A PROGRAM FOR THE DESIGN OF PROCUREMENT SYSTEMS
Pages: 168 S.M.Thesis/May 1976 $8.35
Keywords: expert systems, business applications, procurement systems
Abstract: Computer technology has had a limited success in producing useful business applications. Management systems seldom meet users' requirements, are often inappropriate to an application, and are frequently abandoned. But why?

Business lacks expertise in the application of computers. Managers who are expert in solving business problems find it difficult to specify formal procedures for the solution of these problems. It is not surprising that programmers who work from poorly defined specifications produce poorly written software. The computer industry has provided only limited support in business appications. Application packages are seldom appropriate to a problem and are often misapplied. Misapplication is the principle reason for their failure in practice situations.

Improvements are certainly possible. The manager could be supplied with a system that assists in the design of an application. This system could help the manager specify his requirements by providing him with a framework for thinking about issues relevant to the design of a particular application. The manager might then be better equipped to select a commercial package or to guide in the design of his implementation.

This thesis describes a prototype version of such a system. PROCTOR is a program that assists in the design of a hierarchical planning and control system for a procurement firm.

PROCTOR is implemented as an "unstructured" questionnaire. It guides the user in investigating various aspects of a problem while giving him complete freedom in deciding how and when to supply answers to questions. It allows him to change and skip answers whenever he desires. PROCTOR is implemented in OWL, a system for representing and processing conceptual knowledge. It uses the OWL data base to represent procedures for the questionnaire and to store data accumulated during the iteration. This representation makes possible the presentation of an English-like problem description, various evaluations and the reasons for the evaluations. {AD A026-688}

TR-161
Hack, M.
DECIDABILITY QUESTIONS FOR PETRI NETS
Pages: 194 Ph.D. Dissertation/June 1976 $9.15
Keywords: Petri nets, automata, concurrency
Abstract: An understanding of the mathematical properties of Petri Nets is essential when one wishes to use Petri Nets as an abstract model for concurrent systems. The decidability of various problems which arise in this context is an important aspect of this question. The fact that these problems also arise in the context of other mathematical theories, such as commutative semigroups, closure under linear relations, Matrix Context-Free grammars, or Weak Counter Automata, provides further motivation.

The Reachability Problem for Vector Addition Systems - whose decidability is still an open question - is of central importance. We show that a number of Petri Net problems are recursively equivalent to this problem. These include the Liveness Problem (e.g. can a system reach a deadlocked state?), the single-place reachability problem (can a given buffer ever be emptied?), the persistence problem (can a given transition ever be disabled by the firing of another transition?), and the membership and emptiness problems for certain classes of languages generated by Petri Nets.

The power of the unrestricted Petri Net model is illustrated by various undecidable equivalence results. In particular, we show that the equality of Reachability Sets and the equivalence of two Petri Nets in terms of their language-generating capability are recursively undecidable.

It is hoped that the constructions used to prove our results will shed some light on the source of the complexities of the unrestricted Petri Net model, and may eventually permit us to achieve an optimal balance between representational transparency and analytical power of the Petri Net model.

TR-162
Kent, S.
ENCRYPTION-BASED PROTECTION PROTOCOLS FOR INTERACTIVE USER-COMPUTER COMMUNICATION
Pages: 121 S.M.Thesis/June 1976 $6.95
Keywords: computer security, encryption, cryptology, security, protection
Abstract: This thesis develops a complete set of protocols, which utilize a block cipher, e.g., the NBS data encryption standard, for protecting interactive user-computer communication over physically unsecured channels. The use of the block cipher protects against disclosure of message contents to an intruder, and the protocols provide for the detection of message stream modification and denial of message service by an intruder. The protocols include facilities for key distribution, two-way login authentication, resynchronization following channel disruption, and expedition of high priority messages. The thesis presents designs for modules to implement the protocols, both in the terminal and in a host computer system, and discusses the results of a test implementation of the modules on Multics. {AD A026-911}

TR-163
Montgomery, W.A.
A SECURE AND FLEXIBLE MODEL OF PROCESS INITIATION FOR A COMPUTER UTILITY
Pages: 124 S.M.& E.E. Thesis/June 1976 $7.05
Keywords: protection, computer utility, process initiation, security, Multics
Abstract: This thesis demonstrates that the amount of protected, privileged code related to process initiation in a computer utility can be greatly reduced by making process creation unprivileged. The creation of processes can be controlled by the standard mechanism for controlling entry to a domain, which forces a new process to begin execution at a controlled location. Login of users can thus be accomplished by an unprivileged creation of a process in the potential user's domain, followed by authentication of the user by an unprivileged initial procedure in that domain.

The thesis divides the security constraints provided by a computer utility into three classes: Access control, prevention of unauthorized denial of servic, and confinement. We develop a model that divides process changing, resource control, authentication, and environment initialization. We show which classes of security constraints depend on each of these functions and show how to implement the functions such that these are the only dependencies present.

The thesis discusses an implementation of process initiation for the Multics computer utility based on the model. The major problems encountered in this implementation are presented and discussed. We show that this implementation is substantially simpler and more flexible than that used in the current Multics system.

TR-164
Reed, D.P.
PROCESSOR MULTIPLEXING IN A LAYERED OPERATING SYSTEM
Pages: 207 S.M.Thesis/July 1976 $9.55
Keywords: processor multiplexing, kernel-structured operating system, security kernel, Multics
Abstract: This thesis presents a simply structured design for the implementation of process in a kernel-structured operating system. The design provides a minimal mechanism for the support of two distinct classes of processes found in the computer system - those which are part of the kernel operating system itself, and those used to execute user-specified computations. The design is broken down into two levels, one which implements a fixed number of virtual processors, which are then used to run kernel processes, and are multiplexed to provide processes for user computations. Eventcount primitives are provided, in order to provide a simple unified interprocess control communication mechanism. The design is intended to be used in the creation of a secure kernel for the Multics operating system.

TR-165
McLeod, D.
HIGH LEVEL EXPRESSION OF SEMANTIC INTEGRITY SPECIFICATIONS IN A RELATIONAL DATA BASE SYSTEM
Pages: 119 S.M.Thesis/September 1976 $6.90
Keywords: relational data base, data base, semantics
Abstract: The "semantic integrity" of a data base is said to be violated when the data base ceases to represent a legitimate configuration of the application environment it is intended to model. In the context of the relational data model, it is possible to identify multiple levels of semantic integrity information: (1) the description of the domains of the data base, as abstract sets of atomic data values (domain definition), (2) the specification of the fundamental structure of the relations of the data base (relation structure specification), (3) the definition of the abstract operations which are meaningful in terms of the application environment (structured operations), and (4) the expression of additional semantic information not contained in the structure of the relations nor in the identities of their underlying domains (relation constraints).

A high level, nonprocedural domain definition language facilitates the description of domains. Such a language allows the specification of the properties of the values constituting a domain, and the action that is to occur if an attemp is made to update a column entry such that it does not belong to the underlying domain of that column. The specification of relation structure and structured operations can also be accomplished by means of high level integrity (sub)languages.

A relation constraint has three components: (1) the assertion predicate on the state of the data base or on transitions between data base states), (2) the validity requirement (the occasion(s) at which the assertion must hold), and (3) the violation-action (the action that is to occur if the assertion does not hold at a time when it should). Relation constraint specification can be related to an expression framework (classification scheme) which is useful for the construction of a relation constraint language and specification methodology. Assertions are more than expressions of some relationships among different values in a data base; an assertion singles out the data that is constrained, the states the properties that this data must possess. A classification is provided of the various predicate types used to identify constrained data and to state the properties that they are to possess.

A semantic integrity subsystem (of a generalized relational data base management system) can support the generation and maintenance of integrity specifications, verify that these specifications are met by the data base, and take appropriate action if violations are detected. {AD A034-184}

TR-166
Chan, A.Y.
INDEX SELECTION IN A SELF-ADAPTIVE RELATIONAL DATA BASE MANAGEMENT SYSTEM
Pages: 96 S.M.Thesis/September 1976 $6.20
Keywords: relational data base, data base
Abstract: The development of large integrated data bases that support a variety of applications in an enterprise promises to be one of the most important data processing activities of the next decade. The effective utilization of such data bases depends on the ability of data base management systems to cope with the evolution of data base applications. In this thesis, we attempt to develop a methodology for monitoring the developing pattern of access to a data base and for choosing near-optimal physical data base organizations based on the evidenced mode of use. More specifically, we consider the problem of adaptively selecting the set of secondary indices to be maintained in an integrated relational data base. Stress is placed on the acquisition of an accurate usage model and on the precise estimation of data base characteristics, through the use of access monitoring and the application of forecasting and smoothing techniques. The cost model used to evaluate proposed index sets is realistic and flexible enough to incorporate the overhead costs of index maintenance, creation, and storage. A heuristic algorithm is developed for the selection of a near-optimal index set without an exhaustive enumeration of all possibilities. {AD A034-185}

TR-167
Janson, P.A.
USING TYPE EXTENSION TO ORGANIZE VIRTUAL MEMORY MECHANISMS
Pages: 210 Ph.D. Dissertation/September 1976 $9.65
Keywords: type extension, virtual memory
Abstract: Much effort is currently being devoted to producing systems that are easy to understand, to verify and to develop. The general methodology for designing such a system consists of decomposing it into a structured set of modules so that the modules can be understood, verified and developed individually, and so that the understanding/verification of the system can be derived from the understanding/verification of its modules. While many of the mechanisms in a computer system have been decomposed successfully into a structured set of modules, no technique has been proposed to organize the virtual memory mechanism of a system in such a way.

The present thesis proposes to use type extension for that purpose. The virtual memory mechanism consists of a set of type manager modules implementing abstract information containers. The structure of the mechanism reflects the structure of the containers that are implemented. While using type extension to organize a virtual memory mechanism is conceptually simple, it is hard to achieve in practice. All existing or proposed uses type extension assume the existence of information containers that are uniformly accessible, can always be grown and are protected. Using type extension inside a virtual memory mechanism raises implementation problems since such containers are not implemented. Their implementation is precisely the objective of the virtual memory mechanism. In addition to explaining how type extension can be supported inside a virtual memory mechanism, the thesis demonstrates its use in a case study involving a commercial, general-purpose, time-sharing system. It concludes by providing some insights into the organization of virtual memory mechanisms for time-sharing systems.

TR-168
Pratt, V.R.
SEMANTICAL CONSIDERATIONS ON FLOYD-HOARE LOGIC
Pages: 40 September 1976 $4.55
Keywords: Floyd-Hoare logic, logic, logics of programs, theory of computation
Abstract: This paper deals with logics of programs. The objective is to formalize a notion of program description and to give both plausible (semantic) and effective (syntactic) criteria for the notion of truth of a description. A novel feature of this treatment is the development of the mathematics underlying Floyd-Hoare axiom systems independently of such systems. Other directions that such research might take are also considered. This paper grew out of, and is intended to be usable as, class notes for an introductory semantics course. The three sections of the paper are:

1) A frame work for the logic of programs.

Programs and their partial correctness theories are treated as binary relations on states and formulae respectively. Truth-values are assigned to partial correctness assertions in a plausible (Tarskian) but not directly usable way.

2) Particular Programs.

Effective criteria for truth are established for some programs using the Tarskian criteria as a benchmark. This leads directly to a sound, complete, effective axiom system for the theories of these programs. The difficulties involved in finding such effective criteria for other programs are explored.

3) Variations and extensions of the framework.

Alternatives to binary relations for both programs and theories are speculated on, and their possible roles in semantics are considered. We discuss a hierarchy of varieties of programs and the importance of this hierarchy to the issues of definability and describability. Modal logic is considered as a first-order alternative to Floyd-Hoare logic. We give an appropriate axiom system which is complete for loop-free programs and also puts conventional predicate calculus in a different light by lumping quantifiers with non-logical assignments rather than treating them as logical concepts.

TR-169
Safran, C., Desforges, J.F., Tsichlis, P.N.
DIAGNOSTIC PLANNING AND CANCER MANAGEMENT
Pages: 82 September 1976 $5.80
Keywords: clinical decision making, cancer, medical applications, Hodgkins disease, diagnostics
Abstract: This report describes a computer system for evaluating patients with Hodgkins disease. This system uses decision theoretic techniques to aid in the formulation of a diagnostic plan for the cancer patient. During the process of plan formation, a patient model is constructed and modified to help suggest and evaluate reasonable diagnostic and therapeutic altrnatives. The system also directs sensitivity analyses to determine the effect of error on the decision making process.

TR-170
Furtek, F.C.
THE LOGIC OF SYSTEMS
Pages: 176 Ph.D. Dissertation/December 1976 $8.60
Keywords: Petri nets, logic, information flow, event graphs, marked graphs, state graphs
Abstract: We present a theory about the logical relationships associated with system behavior. The rules governing the behavior of a system are expressed by a Petri net. A set of assumptions about the modeling of a system permit us to separate system behavior into two components, what we refer to as information and control. Information is concerned with choices and how they are resolved. Control is concerned with the fixed, repetitive aspects of behavior - those aspects that are independent of choices.

We develop a concept of information that is nonprobabilistic. It is not inconsistent with Shannon's approach, but simply proceeds from a more basic idea: It deals with possibilities, rather than probabilities. Our approach embodies four common notions about information: (1) information distinguishes between alternatives; (2) it resolves choices; (3) it is transmitted and transformed within a system; (4) it says something about past behavior (memory or postcondition) and something about future behavior (prediction). We can identify those points at which information either enters or leaves a system, and we can trace information as it flows through a system.

The control component of system behavior is determined by a system's control structure, which is an event graph (marked graph). We show how the control structure of a system may be interpreted as the system's space/time framework.

When considered separately, the theories of information and control are of limited applicability. When brought together, they provide a technique for predicting and postdicting behavior.

TR-171
Huber, A.H.
A MULTI-PROCESS DESIGN OF PAGING SYSTEM
Pages: 145 S.M.& E.E. Thesis/December 1976 $7.70
Keywords: paging, computer systems, Multics
Abstract: This thesis presents a design for a paging system that may be used to implement a virtual memory on a large scale, demand paged computer utility. A model for such a computer system with a multi-level, hierarchical memory system is presented. The functional requirements of a paging system for such a model are discussed, with emphasis on the parallelism inherent in the algorithms used to implement the memory management functions.

A complete, multi-process design is presented for the model system. The design incorporates two system processes, each of which manages one level of the multi-level memory, being responsible for the paging system functions for that memory. These processes may execute in parallel with each other and with user processes. The multi-process design is shown to have significant advantages over conventional designs in terms of simplicity, modularity, system security, and system growth and adaptability. An actual test implementation on the Multics system was carried out to validate the proposed design.

TR-172
Mark, W.S.
THE REFORMULATION MODEL OF EXPERTISE
Pages: 254 Ph.D. Dissertation/December 1976 $10.95
Keywords: expert systems
Abstract: This research develops a methodology for implementing a class of expert problem-solving programs which must create models of the problems they are given before they can apply their expert knowledge to those problems. That is, given a problem in their domain of expertise, these programs "reformulate" the problem description in terms of their built-in abstract models of the domain. The implementation methodology described in this report provides the following capabilities:
• pieces of expert knowledge can be added to the program in a fairly independent manner without regard to their control structure implications
• programs can delve through irrelevant information in the problem description to pick out only the facts needed for problem-solving
• ograms can model the input problem at various levels of detail; as required by the problem-solving task
• programs can tailor their expert models to the problem at hand

This is accomplished by controlling the general procedure which maps pieces of the expert knowledge base into pieces of the input with feedback information taken from the model of the input so far.

The methodology is demonstrated with a working program which acts as an expert business consultant dealing with firms which have work force control problems. Input is in the form of several paragraphs of simplified English which the program models in accordance with a cause-effect flow model of the firm. If the modeling effort is successful, policy suggestions are made which improve the performance of the firm in question. {AD A035-397}

TR-173
Goodman, N.
COORDINATION OF PARALLEL PROCESSES IN THE ACTOR MODEL OF COMPUTATION
Pages: 116 S.M. Thesis/December 1976 $6.80
Keywords: parallel processing, mutual exclusion problem
Abstract: Two algorithms for the mutual exclusion problem are described and proven to operate correctly. The algorithms are unique in that they use very simple synchronization primitives yet are fair and retain their fairnes even if the number of parallel processes in the computer system increases unbounded over time. One of the algorithms uses simple cells of read/write storage as the primitive; the algorithm is similar to the classic algorithms for this problem proposed by Dijkstra and Knuth, but is generalized to handle an arbitrary number of processes. The second algorithm uses extended cells of storage that model read/modify/write (e.g. test-and-set) instructions. While it is well known how to use read/modify/write instructions to achieve unfair mutual exclusion, their use in a fair algorithm is novel.

The results prove that cells of read/write storage are sufficiently powerful primitives to achieve coordination of parallel processes. There is no theoretical necessity for a model of computation to include more sophisticated synchronization primitives such as semaphores and serializes. But while cells are sufficient, the algorithms are very inefficient; more sophisticated primitives are desirable for that reason.

TR-174
Hunt, D.H.
A CASE STUDY OF INTERMODULE DEPENDENCIES IN A VIRTUAL MEMORY SUBSYSTEM
Pages: 121 S.M. & E.E. Thesis/December 1976 $6.95
Keywords: virtual memory, intermodule dependencies, Multics
Abstract: A problem currently confronting computer scientists is to develop a method for the production of large software systems that are easy to understand and certify. The most promising methods involve decomposing a system into small modules in such a way that there are few intermodule dependencies. In contrast to previous research, this thesis focuses on the nature of the intermediate module dependencies, with the goal of identifying and eliminating those that are found to be unnecessary. Using a virtual memory subsystem as a case study, the thesis describes a structure in which apparent dependencies can be eliminated. Owing to the nature of virtual memory subsystems, many higher level functions can be performed by lower level modules that exhibit minimal interaction. The structuring methods used in this thesis, inspired by the structure of the LISP world of atomic objects, depend on the observation that a subsystem can maintain a copy of the name of an object without being dependent upon the object manager. Since the case study virtual memory subsystem is similar to that of the Multics system, the results reported here should aid in the design of similar sophisticated virtual memory subsystems in the future.

TR-175
Goldberg, H.J.
A ROBUST ENVIRONMENT FOR PROGRAM DEVELOPMENT
Pages: 101 S.M. Thesis/February 1977 $6.35
Keywords: debugging, programming environment, security kernel
Abstract: This thesis examines the problems of debugging and preservation of the user programming environment and proposes a scheme by which the program development environment can be protected.

Typically, designers of timeshared or multiprogrammed computer systems only consider inter-user interference as a source of problems and do not worry about what users do in their own environments. Thus, users can, by writing incorrect programs, cause the destruction of the programming environment and personal data bases. A protection scheme is proposed that satisfies the needs of the user by employing a protection mechanism, rings, that allows the program development environment to be protected from user written programs and yet be outside of the supervisor. Having these programs outside the supervisor satisfies the goals of creating a "security kernel", which is a supervisor containing only security related programs.

This thesis presents a model of the user environment wherein the concept of a "procedural package" is explained. The procedural package contains not only the code for the procedure, but in addition, environment components necessary for the proper execution of the procedure such as dynamic, static, and allocate/free storage. The thesis describes the "inter-procedure interference" problem in terms of the model and proposes an ideal solution based on a domain architecture. Problems with the ideal solution are presented and an alternate solution suggested.

In addition, the thesis identifies and discusses, in detail, environments that are needed to control a user's process, and examines error signalling mechanisms, particularly in their use in an environment like the one proposed to solve the inter-procedure interference problem.

TR-176
Swartout, W.R.
A DIGITALIS THERAPY ADVISOR WITH EXPLANATIONS
Pages: 96 S.M.Thesis/February 1977 $6.20
Keywords: explanation, digitalis, clinical decision making, medical applications
Abstract: This thesis describes the English explanation facility of the OWL Digitalis Advisor, a program designed to advise physicians regarding digitalis therapy. The program is written in OWL, an English-based computer language being developed at MIT. The system can explain, in English, both the methods it uses and how those methods were applied during a particular session. In addition, the program can explain how it acquires information and tell the user how it deals with that information either in general or during a particular session.

Most explanations are produced directly from the code used in prescribing digitalis and from information which is generated by the OWL interpreter as it runs. The ability of the program to translate its internal structure to an English explanation is provided by structuring the program using Semantic Model Programming. Each OWL procedure attempts to represent a single concept or idea that should be meaningful to the physician using the system. By organizing the program in this way, the explanations produced by the system tend to relate well to ideas with which the physician is already acquainted.

In many current systems which ask the user a series of questions, a problem occurs if the user wishes to change his answer to a previous question. These systems accept the change, but must recompute all the results computed subsequent to that question to insure that none of them are affected. Clearly, this may involve a considerable amount of unnecessary recomputation. By using OWL, we obtain the data structures necessary to avoid this problem. An algorithm is described that allows the system to accept a changed answer without recomputing all prior results. This process is called updating. The updating algorithm presented here also allows the system to provide concise explanations of the effects of the changed answer.

TR-177
Mason, A.H.
A LAYERED VIRTUAL MEMORY MANAGER
Pages: 133 S.M.& E.E.Thesis/May 1977 $7.30
Keywords: virtual memory, Multics, layered virtual memory
Abstract: This thesis presents a specification for the Multics virtual memory manager. The virtual memory manager is that part of the operating system which coordinates the usage of physical memory and which manages the bindings between logical memory and physical memory. In the case of Multics, physical memory is composed of fixed-length blocks called frames and logical memory consists of segments, representing sets of frames.

The original specification is out of date and obsolete because it describes an overly complicated structure and ignores the issue of resource control. The specification described here compatibly updates the functionality of the Multics virtual memory manager, simplifies the requisite structure, and addresses resource control problems.

The specification is in the form of a model, using the methodologies of type extension and layers of abstraction. These methodologies provide the tools to develop a precise model structure, which is capable of handling the intricacies of resource control. The end result is organizational simplicity, certifiability, and comprehension.

TR-178
Bishop, P.B.
COMPUTER SYSTEMS WITH A VERY LARGE ADDRESS SPACE AND GARBAGE COLLECTION
Pages: 276 Ph.D. Dissertation/May 1977 $11.60
Keywords: computer systems, address space, garbage collection
Abstract: The concept of objects is beginning to gain acceptance throughout the field of computer science. A new computer system is proposed that provides hardware support for objects and object references that can be used in all applications of objects. The new system provides small object references that can be copied freely, makes very small objects efficient, and retrieves the storage for inaccessible objects automatically. This system is compared with some widely used existing systems, and while its speed seems to be competitive, it is much easier to use.

Object references provide protection in the new system as do capabilities in capability systems. The object reference in the new system contains an address from a linear, paged virtual address space rather than a unique ID. Use of small objects is make feasible by efficiently grouping objects into areas. Objects in the same area may be placed on the same page. The system automatically and efficiently maintains lists of inter-area links that allow single areas to be garbage collected independently of the rest of the system. The garbage collector can determine whether objects have been inappropriately placed in an area and can move these objects to more appropriate areas automatically. {AD A040-601}

TR-179
Karger, P.A.
NON-DISCRETIONARY ACCESS CONTROL FOR DECENTRALIZED COMPUTING SYSTEMS
Pages: 139 S.M. Thesis/May 1977 $7.50
Keywords: decentralized computing systems, computer systems, access control, security, protection

Abstract: This thesis examines the issues relating to non-discretionary access controls for decentralized computing systems. Decentralization changes the basic character of a computing system form a set of processes referencing a data base to a set of processes sending and receiving messages. Because messages must be acknowledged, operations that were read-only in a centralized system become read-write operations. As a result, the lattice model of non-discretionary access control, which mediates operations based on read versus read-write considerations, does not allow direct transfer of algorithms from centralized systems to decentralized systems. This thesis develops new mechanisms that comply with the lattice model and provide the necessary functions for effective decentralized computation.

Secure protocols at several different levels are presented in the thesis. At the lowest level, a host to host protocol is shown that allows communication between hosts with effective internal security controls. Above this level, a host independent naming scheme is presented that allows generic naming of services in a manner consistent with the lattice model. The use of decentralized processing to aid in the downgrading of information is shown in the design of a secure intelligent terminal. Schemes are presented to deal with the decentralized administration of the lattice model, and with the proliferation of access classes as the user community of a decentralized system become more diverse. Limitations in the use of end-to-end encryption when used with the lattice model are identified, and a scheme is presented to relax these limitations for boradcast networks. Finally, a scheme is presented for forwarding authentication information between hosts on a network, without transmitting passwords (or their equivalent) over a network. ✓AD A040-804}

TR-180
Luniewski, A.
A SIMPLE AND FLEXIBLE SYSTEM INITIALIZATION MECHANISM
Pages: 105 S.M. & E.E. Thesis/May 1977 $6.50
Keywords: system initialization, layered systems
Abstract: This thesis presents an approach to system initialization which is simple and easy to understand and, at the same time, is versatile in the face of configuration changes. This thesis considers initialization of a layered system. The initialization mechanism is built upon three key concepts: existence of a minimal configuration, a core image of the system and dynamic reconfiguration. By assuming that the system will be running on the minimal configuration we generate a core image of the base layer of the system which, when loaded into core on any viable configuration, produces an operable base layer. As higher layers of the system are initialized dynamic reconfigurations of the lower layers are invoked to cause the system to run on the configuration actually present. The thesis also considers the problems one might encounter in implementing the many dynamic reconfigurations required by this approach to system initialization.

TR-181
Mayr, E.W.
THE COMPLEXITY OF THE FINITE CONTAINMENT PROBLEM FOR PETRI NETS
Pages: 65 S.M. Thesis/June 1977 $5.30
Keywords: Petri nets, finite containment
Abstract: If the reachability set of a Petri net (or, equivalently, vector addition system) is finite it can be effectively constructed. Furthermore, the finiteness is decidable. Thus, the containment and equality problem for finite reachability sets become solvable. We investigate the complexity of decision procedures for these two problems and show by reducing a bounded version of Hilbert's Tenth Problem to the finite containment problem that these two problems are extremely hard, that, in fact, the complexity of each decision procedure exceeds any primitive recursive function infinitely often. The finite containment and equality problem are thus the first uncontrived, decidable problems with provably non-primitive recursive complexity.

TR-182
Brown, G.P.
A FRAMEWORK FOR PROCESSING DIALOGUE
Pages: 73 June 1977 $5.50
Keywords: processing dialogue, linguistics
Abstract: This report describes a framework for handling mixed-initiative English dialogue in a console session environment, with emphasis on recognition. Within this framework, both linguistic and non-linguistic activities are modeled by structures called *methods,* which are a declarative form of procedural knowledge. Our design focuses on units of linguistic activity larger than the speech act, so that the pragmatic and semantic context of an utterance can be used to guide its interpretation. Also important is the treatment of indirect speech acts, e.g., the different ways to ask a question, give a command, etc.

Given the static model of dialogue embodied in the methods, the problem is to find the correct method step that relates to a particular input. We handle this problem through a combination of careful structural distinctions and the use of multiple recognition strategies. The dialogue methods are used to generate expectations dynamically, special structures are used to facilitate matching, and a basic distinction between four major utterance classes is used to determine which of several overall matching strategies should be used for a given expectation. {AD A042-370}

TR-183
Jaffe, J.M.
SEMILINEAR SETS AND APPLICATIONS
Pages: 79 S.M.Thesis/July 1977 $5.70
Keywords: semilinear sets, theory of computation, vector addition
Abstract: We study semilinear sets as they arise in several areas of automaton and formal language theory and logic. In particular we develop the properties of an important subclass, the slice.

We show that any semilinear set is the projection of some slice. An application of this idea provides our main result which is a new proof of Parikh's theorem concerning letter counts in context-free languages. It is shown that when letter counts and derivations are coded in a natural way as vectors of integers, then the set of codes associated with any context-free language is a slice.

The application of semilinear sets to vector addition systems and logical languages are developed. Certain models are shown to be equivalent, but differences between models are also explored.

TR-184
Levine, P.H.
FACILITATING INTERPROCESS COMMUNICATION IN A HETEROGENEOUS NETWORK ENVIRONMENT
Pages: 105 S.B. & S.M. Thesis/July 1977 $6.50
Keywords: interprocess communication, networks, computer networks
Abstract: Passing information among processors with different internal data formatting schemes has proven to be a major complication to computer networking efforts. Data format translation is necessary to support information exchange in a heterogeneous network environment. Three strategies for performing this translation for communications between a message sender and receiver are: translation by the receiver, translation by an intermediate translator, and the use of a standard intermediate format. The standard format is shown to be the most responsive to a set of general network design principles. .br The implementation of an intermediate format based interprocess communications scheme requires a mechanism for passing the semantic description of each string of data bits. Two alternative mechanisms for passing this information are discussed, and data "tagging" is selected as the more flexible. Other implementation considerations are examined, including possible problems in performing translator and the relationship formal translation has to other network message handling functions. {AD A043-901}

TR-185
Goldman, B.
DEADLOCK DETECTION IN COMPUTER NETWORKS
Pages: 180 S.B.& S.M.Thesis/September 1977 $8.75
Keywords: deadlock detection, computer networks
Abstract: In this thesis, two published algorithms dealing with deadlock detection in computer networks are discussed and examples demonstrating the failure of these algorithms are given. {AD A047-025}

TR-186
Ackerman, W.B.
A STRUCTURE MEMORY FOR DATA FLOW COMPUTERS
Pages: 125 S.M. Thesis/August 1977 $7.10
Keywords: data flow, structure memory, data flow computers, concurrency, packet communication
Abstract: A data flow computer is one which achieves enormous concurrency of instruction execution through a machine architecture that acts directly on a data dependency graph of the program. To handle arrays and data structures effectively, a data flow computer must have access to a memory system which can handle large numbers of concurrent transactions. This thesis presents a design for such a memory. A "cache"

mechanism is presented for improving the performance of the system, and a mechanism is given for using sequential-access devices such as shift registers as the memory medium. The memory system design uses the "packet communication" concept, in which the components of the system communicate only through the transmission of fixed size "packets" of data. {AD A047-026}

TR-187
Long, W.J.
A PROGRAM WRITER
Pages: 279 Ph.D. Dissertation/November 1977 $11.70
Keywords: automatic programming
Abstract: This thesis is concerned with the problem of taking a high level specification for a program and designing an appropriate algorithm and data structure, utilizing knowledge about the domain and about programming. The basic approach in this thesis is to use successive refinement organized and regulated by several models of different aspects of the programs. The system, called the program writer, uses five models of the programs: as representing events in the application domain, as primitives passing arguments within a control structure, as creating and using data, as carrying on an I/O exchange with the user, and as a construct in the target language. These models provide the appropriate views of the program to constrain and guide the refinement process. The global views provided by these models also make possible global transformations of the program structure when an opportunity for improvement is recognized. {AD A047-595}

TR-188
Bryant, R.E.
SIMULATION OF PACKET COMMUNICATION ARCHITECTURE COMPUTER SYSTEMS
Pages: 120 S.M.Thesis/November 1977 $6.95
Keywords: packet communication, computer systems, concurrency, architecture
Abstract: Simulations of computer systems have traditionally been performed on a single sequential computer, even if the system to be simulated contains a number of components which operate concurrently. An alternative would be to simulate these systems on a network of processors. With this approach, each processor would simulate one component of the system, hence the component simulations would proceed concurrently. By exploiting the modularity and concurrency in the system to be simulated, the simulation would itself be modular and concurrent.

An accurate simulation must model the time behavior of the system as well as its input-output behavior. In order to avoid real-time constraints on the processors and communication network in the simulation facility, the simulation timing must use a time-independent algorithm. That is, the simulated behavior of each component should not depend ont he speed at which the simulation is performed.

With this time-independent approach, additional coordination operations are required to prevent deadlock of the simulation. This coordination can be provided without any centralized control. Instead, the program for the simulation of each component is modified t simulations. Additional termination operations are also required to assure that the simulation will terminate under the exact same conditions that the system being simulated would terminate. These operations can also be provided without any centralization of control or real-time constraints. Furthermore a simulation which uses these coordination and termination operations is provably correct. That is, the simulation will accurately model both the time behavior and the input-output behavior of the system. {AD A048-290}

TR-189
Ellis, D.J.
FORMAL SPECIFICATIONS FOR PACKET COMMUNICATION SYSTEMS
Pages: 138 Ph.D. Dissertation/November 1977 $7.45
Keywords: packet communication, computer systems
Abstract: One of the most difficult tasks facing computer scientists is that of designing systems and making sure that they perform their intended functions correctly. As computer systems have grown in size and complexity, the problems of system design and verification have become increasingly acute. Formal specifications, which are precise descriptions of a system's function, provide a basis for understanding system operation as well as for proving correctness.

Although there has been much work in formal specification and verification of computer *programs*, relatively little research has been done on *system* specification. A particular class of asynchronous systems, known as *packet communication systems*, has been chosen as the subject of this study. Packet communication systems are composed of independently operating units that interact only by transmitting *packets* of information. These systems possess a number of desirable structuring properties that make them suitable for formal analysis.

We have developed a model for formally describing the behavior of packet systems and for proving correctness. The model is based on the fact that packet systems may be viewed both *externally*, in terms of their interaction with the outside world, and *internally*, in terms of their structural composition from smaller units. A packet system is shown to be correct by proving that its formal characterizations corresponding to these two views are equivalent.

Our model is used to prove the correctness of three sample packet systems, and a general characterization of acyclic systems is stated and proved. {AD A048-380}

TR-190
Moss, E.B.
ABSTRACT DATA TYPES IN STACK BASED LANGUAGES
Pages: 154 S.M.Thesis/February 1978 $7.95
Keywords: stack based languages, abstract data types, CLU, garbage collection
Abstract: Abstract data types are the basis of an emerging methodology of computer programming. The only existing languages supporting abstract data types directly, CLU and Simula, both require compacting garbage collection, and thus they are not suitable for many applications. This thesis presents the design of a new language incorporating abstract data types; the language requires only a run-time stack, and not garbage collection. This new language, called ASBAL (for "A Stack Based Abstraction Language"), is based on CLU, and borrows as many features as possible directly from it. Virtually every significant feature of CLU is carried over into ASBAL in some form, and extensions are included when necessary. For example, the maximum size of objects becomes an issue and is resolved by the addition of size parameters to types. Also, a limited facility for dynamic storage allocation is incorporated in ASBAL to compensate for the removal of a garbage collected heap. This facility allows list and graph structures to be built within the framework of the stack while preventing dangling references as a "side-effect" of compile-time type checking. {AD A052-332}

TR-191
Yonezawa, A.
SPECIFICATION AND VERIFICATION TECHNIQUES FOR PARALLEL PROGRAMS BASED ON MESSAGE PASSING SEMANTICS
Pages: 221 Ph.D. Dissertation/January 1978 $9.95
Keywords: parrallelism, message passing
Abstract: This thesis presents formal specification and verification techniques for both serial and parallel programs written in SIMULA-like object oriented languages.

These techniques are based on the notion of states of individual objects which are defined uniformly in serial and parallel computations. They can specify and verify the behavior of data and procedural objects in multi-process environments, thus overcoming some of the difficulties in dealing with parallelism which characterized previous work on formal specifications for abstract data types. Among others, the specifications and verifications of a bounded buffer and air line reservation systems are given.

Using a model of a simple post office, we illustrate our specification and verification techniques for systems, such as operating systems and multi-user data base systems, which are characterized by complex internal concurrent activities. It it demonstrated that the specifications of the overall functions of the system which we call task specifications can be derived from specifications of the individual behavior and mutual interaction of the subsystems.

A method of defining states of individual objects as mathematical functions is suggested. {AD A051-149}

TR-192
Niamir, B.
ATTRIBUTE PARTITIONING IN A SELF-ADAPTIVE RELATIONAL DATA BASE SYSTEM
Pages: 157 S.M. Thesis/January 1978 $8.05
Keywords: data base management, attribute partitioning, file systems, transaction processing

Abstract: One technique that is sometimes employed to enhance the performance of a data base management system is known as attribute partitioning. This is the process of dividing the attributes of a file into subfiles that are stored separately. By storing together those attributes that are frequently requested together by transactions, and by separating those that are not, attribute partitioning can reduce the number of pages that must be transferred from secondary storage to primary memory in order to process a transaction.

The goal of this work is to design mechanisms that can automatically select a near-optimal attribute partition of a file's attributes, based on the usage pattern of the file and on the characteristics of the data in the file. The approach taken to this problem is the large space of possible partitions. The heuristics propose a small set of promising partitions to submit for detailed analysis. The estimator assigns a figure of merit to any proposed partition that reflects the cost that would be incurred in processing the transactions in the usage pattern if the file were partitioned in the proposed way. We have also conducted an extensive series of experiments with a variety of design heuristics; as a result, we have identified a heuristic that nearly always finds the optimal partition of a file.

The context of this study is a relational data base management system that can process transactions made against relations whose physical partitioning is unknown to the user. In specifying and modeling this system, it is necessary to address the problem of optimizing nonprocedural queries made to a partitioned file. We have derived a number of such optimization techniques and have provided the results of a number of experiments with them. {AD A053-292}

TR-193
Schaffert, C.
A FORMAL DEFINITION OF CLU
Pages: 179 S.M. Thesis/January 1978 $8.70
Keywords: CLU, programming languages, programming methodology
Abstract: This thesis develops a new language definition methodology that overcomes certain limitations of existing techniques, including the problems arising from shared variables. The methodology is then used to provide a complete formal definition of the programming language CLU. CLU is an object-oriented language which was designed to incorporate the idea of data abstraction. CLU contains a coroutine-like mechanism for general iteration, and a powerful exception handling mechanism. These features are typical of those found in advanced programming languages, and the fact that they are difficult to handle in existing language definition methods motivated the current work. The CLU definition is presented as a means of evaluating the new technique.

The proposed method is a synthesis of translational and axiomatic techniques. To define a language, we give rules for translating any procedure in the language into a mathematical structure. This structure directly describes the computations that the procedure should perform. However, the structure is so designed that an axiomatic description of the procedure can be derived from it. Thus our method can have the advantages of both axiomatic and translational methods. In addition, both the language designer and the programmer can use the dual nature of the method to double check their work.

TR-194
Hewitt, C., Baker, H.G.
ACTORS AND CONTINUOUS FUNCTIONALS
Pages: 29 February 1978 $4.20
Keywords: parallelism, concurrency, actors, distributed computer systems, multiprocessing, networks, interprocess communication, message passing systems
Abstract: This paper presents precise versions of some "laws" that must be satisfied by computations involving communicating parallel processes. The laws take the form of stating plausible restrictions on the histories of computations that are physically realizable. The laws are very general in that they are obeyed by parallel processes executing on a time varying number of distributed physical processors. For example, some of the processors might be in orbiting satellites. The laws are justified by appeal to physical intuition and are to be regarded as falsifiable assertions about the kinds of computations that occur in nature rather than as proved theorems in mathematics. The laws are intended to be used to analyze the mechanisms by which multiple processes can communicate to work effectively together to solve difficult problems.

The laws presented in this paper are intended to be applied to the design and analysis of systems consisting of large numbers of physical processors. The development of such systems is becoming economical because of rapid progress in the development of large scale integrated circuits.

We generalize the usual notion of the history of a computation as a sequence for events for the notion of a partial order of events. Partial orders of events seem better suited to expressing the causality involved in parallel computations than totally ordered sequences of events obtained by "considering all shuffles" of the elementary steps of the various parallel processes [21, 22]. The utility of partial orders is demonstrated by using them to express our laws for distributed computation. These laws in turn can be used to prove the usual induction rules for proving properties of procedures. They can also be used to derive the continuity criterion for graphs of functions studied in the Scott-Strachey model of computation. The graph of a function is simply the set of all input output pairs for the function. We can prove that the graph of any physically realizable procedure **p** that behaves like a mathematical function is the limit of a continuous functional **F** such that.

$$\text{graph}(\mathbf{p}) = \cup_{i \in \mathbf{N}} \mathbf{F}^{i}(\{\})$$

In other words the graph of **p** is the limit of the n-fold compositions of **F** with itself beginning with the empty graph. {AD A052-266}

TR-195
Bruss, A.R.
ON TIME-SPACE CLASSES AND THEIR RELATION TO THE THEORY OF REAL ADDITION
Pages: 31 S.M.Thesis/March 1978 $4.25
Keywords: computational complexity, theory of real addition, time and space, log-space reducibility
Abstract: A new lower bound on the computational complexity of the theory of real addition and several related theories is established: any decision procedure for these theories requires either space $2^{\in n}$ or nondeterministic time $2^{\in n^2}$ for some constant $\in > 0$ 0 and infinitely many n.

The proof is based on the families of languages TISP(T(n),S(n)) which can be recognized simultaneously in time T(n) and S(n) and the conditions under which they form a hierarchy.

TR-196
Schroeder, M.D., Clark, D.D., Saltzer, J.H., Wells, D.
FINAL REPORT OF THE MULTICS KERNEL DESIGN PROJECT
Pages: 111 March 1978 $6.65
Keywords: protection, security, security kernel, Multics, type extension, operating systems, supervisors, verification
Abstract: We describe a plan to create an auditable version of Multics. The engineering experiments of that plan are now complete. Type extension as a design discipline has been demonstrated feasible, even for the internal workings of an operating system, where many subtle intermodule dependencies were discovered and controlled. Insight was gained into several tradeoffs between kernel complexity and user semantics. the performance and size effects of this work are encouraging. We conclude that verifiable operating system kernels may someday be feasible.

TR-197
Baker, H.G.
ACTOR SYSTEMS FOR REAL-TIME COMPUTATION
Pages: 145 Ph.D. Dissertation/March 1978 $7.70
Keywords: real-time systems, parallelism, message passing, semantics of parallelism, storage management, garbage collection, list memory, continuations, models for distributed computation, partial orders of events
Abstract: Actor theory was invented by Hewitt and collaborators as a synthesis of many of the ideas from the high-level languages LISP, GEDANKEN, SMALLTALK, SIMULA-67, and others. Actor theory consists of a group of active objects called *Actors*, which communicate by passing messages to one another.

This thesis explores several problems associated with implementing Actor theory as a basis for computer systems design. First, we give a firmer foundation to the theory by setting forth axioms which must be satisfied by any physically realizable message passing system. We then give an operational semantics for this theory by exhibiting an interpreter which is mapping this conceptual system onto current hardware in such a way that simple primitive operations all take a (small) bounded amount of time. In particular, the issues of storage and processor management are investigated and a real-time incremental garbage collection system for both is exhibited and analyzed. {AD A053-328}

TR-198
Halstead, R.H.
MULTIPLE-PROCESSOR IMPLEMENTATIONS OF MESSAGE PASSING SYSTEMS
Pages: 172 S.M.Thesis/April 1978 $8.50
Keywords: message passing, distributed computing, actors, networks
Abstract: The goal of this thesis is to develop a methodology for building networks of small computers capable of the same tasks now performed by single larger computers. Such networks promise to be both easier to scale and more economical in many instances.

The μ calculus, a simple syntactic formalism for representing message passing computations, is presented and augmented to serve as the semantic basis for programs running on the network. The augmented version includes cells, tokens, and semaphores, as well as primitives for side-effect-free computation. Tokens, a novel construct, allow certain simple communications and synchronization tasks without involving fully general side effects.

The network implementation presented supports object references, keeping track of them by using a new concept, the reference tree. A reference tree is a group of neighboring processors in the network that share knowledge of a common object. Also discussed are mechanisms for handling side effects on objects and strategy issues involved in allocating computations to processors. {AD A054-009}

TR-199
Terman, C.J.
THE SPECIFICATION OF CODE GENERATION ALGORITHMS
Pages: 86 S.M.Thesis/April 1978 $5.90
Keywords: machine-independent code generation, compiler meta-languages
Abstract: This thesis addresses the problem of automatically constructing the code generation phrase of a compiler from a specification of the source language and target machine. A framework for such a specification is presented in which information about language- and machine-dependent semantics is incorporated as a set of transformations on an internal representation, and the metalanguage in which the transformations are written are discussed in detail.

The major goal of this approach is to separate machine-and language-dependent knowledge (as embodied in a transformation catalogue) from general knowledge about code generation. This general knowledge is supplied by the third component of the framework: a meta-interpreter is also capable of selecting and applying transformations from the transformation catalogue. The three component framework described in the thesis provides a specification that can easily be tailored to new languages and machine architectures without compromising the ability to generate optimal code. {AD A054-301}

TR-200
Harel, D.
LOGICS OF PROGRAMS: AXIOMATICS AND DESCRIPTIVE POWER
Pages: 152 Ph.D. Dissertation/May 1978 $7.90
Keywords: arithmetical, axiom system, computation tree, divergence, dynamic logic, execution method, failure, first order logic, guarded commands, logics of programs, propositional dynamic logic, recursive programs, regular programs, relative completeness, total correctness, weakest preconditions
Abstract: This thesis is concerned with the development of mathematical tools for reasoning about computer programs. The approach is to design and investigate the properties of various *dynamic logics* with an emphasis on useful expressive power and adequate proof theory.

First, rigorous definitions of the propositional and first-order dynamic logics are given, with an emphasis on the flexibility obtained by leaving unspecified the class of programs which these logics can discuss. A large portion of the result obtained to date in the investigation of dynamic logic is included and put in proper perspective. Then, a proof theory is developed based upon the idea of axiomatizing the first order dynamic logics relative to *arithmetical universes*. Such axiomatizations are supplied and proved *arithmetically complete* for the regular (flowcharts) and context-free (recursive programs) cases. The notions of *diverging* and *failing* are then introduced, with the aid of which the concept of the *total correctness* of a nondeterministic program is defined and the concept of a *weakest precondition* clarified. A detailed investigation of the properties of diverging and failing is then carried out, including the construction of arithmetically complete axiomatizations of both the regular and context-free logics obtained by supplying dynamic logic with the ability to discuss diverging directly.

If a termination condition, a list of zero or more objects, and a new environment; the termination condition is used to govern control flow. This uniform treatment of expressions and statements allows a simple definition of the run-time exception handling mechanism provided in CLU. The meaning of a procedure generator or iterator generator is a function that takes a list of actual parameters, a list of arguments, and an environment, and produces a result as for statement and expression evaluation. The meaning of parameters is given in terms of textual substitution. A non-parameterized routine is viewed as being a generator with an empty parameter list. The meaning of a cluster is a function that takes a list of actual cluster parameters and an operation name, and produces the meaning of that operation.

TR-201
Scheifler, R.
A DENOTATIONAL SEMANTICS OF CLU
Pages: 175 S.M. Thesis/May 1978 $8.60
Keywords: denotational semantics, programming language semantics, CLU, Scott-Strachey method
Abstract: A denotational semantics of CLU, an object-oriented language supporting data abstractions, is presented. The definition is based on Scott's lattice-theoretic approach to the theory of computation. Modules, the basic unit of compilation, are represented in terms of a set of recursively defined domains called the abstract syntax. As part of checking the legality of a module, a transformation is made from the abstract syntax to a modified syntax; the transformation reflects the results of compile-time computations that need not be repeated at run-time. An execution environment is defined to be a set of objects, each with a particular state, together with a mapping from variable names to objects. The meaning of an expression or statement is a function that takes an environment and produces a result consisting of a termination condition, a list of zero or more objects, and a new environment, the termination condition is used to govern control flow. This uniform treatment of expressions and statements allows a simple definition of the run-time exception handling mechanism provided in CLU. The meaning of a procedure generator or iterator generator is a function that takes a list of actual parameters, a list of arguments, and an environment, and produces a result as for statement and expression evaluation. The meaning of parameters is given in terms of textual substitution. A non-parameterized routine is viewed as being a generator with an empty parameter list. The meaning of a cluster is a function that takes a list of actual cluster parameters and an operation name, and produces the meaning of that operation.

TR-202
Principato, R.N.
A FORMALIZATION OF THE STATE MACHINE SPECIFICATION TECHNIQUE
Pages: 102 S.M.& E.E.Thesis/July 1978 $6.40
Keywords: state machine specifications, data abstractions, formal specification technique, proofs of correctness
Abstract: This thesis develops the state machine specification technique, a formal specification technique for data abstractions based on Parnas' work on specifying software modules. When using the state machine technique, each data object is viewed as the state of an abstract (and not necessarily finite) state machine and, in the state machine, this state set is implicitly defined. The basic idea is to separate the operations of the data abstraction into two distinct groups; those which do not change the state but allow some aspect of the state to be observed, the value returning of V-functions, and those which change the state, the operation or O-functions. The specifications are then written by stating the effect of each O-function on the result of each V-function. This implicitly defines the smallest set of states necessary to distinguish the variations in the results of the V-functions. It also determines the transitions among these states caused by the O-functions.

An abstract model for the semantics of state machine specifications is presented and then used to formalize the semantics of a concrete specification language. Furthermore, a methodology for proving the correctness of an implementation of a data abstraction specified by a state machine is discussed and illustrated.

TR-203
Laventhal, M.S.
SYNTHESIS OF SYNCHRONIZATION CODE FOR DATA ABSTRACTIONS
Pages: 228 Ph.D. Dissertation/July 1978 $10.15
Keywords: synchronization, synthesis, data abstractions, abstract data types, concurrency, interprocess communication, monitors, deadlock, starvation

Abstract: Synchronization code is necessary to control shared access of an abstract data object in a parallel-processing environment. This thesis explores an approach in which a synchronization property can be specified in a high-level nonprocedural language and an implementation for the specified property can be synthesized algorithmically. A *problem specification* language is introduced in which synchronization properties can be expressed in a structured but natural manner. A method is then presented for synthesizing an implementation. An intermediate form, called a *solution specification*, is first derived, representing an abstract solution to the problem. The derivation of the solution specification accomplishes the transformation of the specification from nonprocedural to procedural from. The solution specification can be translated directly into a source language synchronization mechanism, such as a monitor.

Specifications for common synchronization properties, such as the readers-writers and bounded buffer problems, are expressed in the problem specification language. Corresponding implementations are then synthesized for these problems. In addition, the derived solution specification can be used in analyzing the soundness of the original problem specification with respect to criteria such as freedom from deadlock and starvation. {AD A058-232}

TR-204
Teixeira, T.J.
REAL-TIME CONTROL STRUCTURES FOR BLOCK DIAGRAM SCHEMATA
Pages: 79 S.M. Thesis/August 1978 $5.70
Keywords: real-time scheduling, priority scheduling, deadline-driven scheduling, control structures
Abstract: Block diagram schemata model computation systems in the context of an external environment. The environment imposes various constraints on the real-time performance of any implementation of a block diagram schema. The model is used to provide precise definitions of real-time performance. The portion of the implementation that affects the real-time performance is called the control structure.

This research investigates several strategies for synthesizing control structures to satisfy the external real-time specifications. The simplest strategy is to execute all the blocks in the diagram in some fixed order. Control structures of this type have been somewhat ignored for time critical applications. The synthesis problem is shown to be solvable in the sense that acyclic control structures do not need to be considered. A branch-and-bound synthesis algorithm is presented which requires exponential time in the worst case. Although no efficient synthesis algorithm was found, the conjecture that the problem is NP-complete is not proved.

The other strategy for implementing control structures makes use of the fact that in some applications the input values change at discrete times. Under this assumption, block diagram schemata are similar to traditional models of real-time tasks is presented that guarantees the real-time specifications will be met. This algorithm relaxes previous restrictions of the deadline for a task being coincident with its next request.

Finally, some of the issues involved with multiple processor control structures are discussed, although no specific algorithms are investigated. {AD A061-122}

TR-205
Reed, D.P.
NAMING AND SYNCHRONIZATION IN A DECENTRALIZED COMPUTER SYSTEMS
Pages: 181 Ph.D. Dissertation/October 1978 $8.75
Keywords: distributed computer systems, reliability, synchronization
Abstract: In this dissertation a new approach to the synchronization of accesses to shared data objects is developed. Traditional approaches to the synchronization problems of shared data accessed by concurrently running computations have relied on mutual exclusion -- the ability of one computation to stop the execution of other computations that might access or change shared data accessed by that computation. Our approach is quite different. We regard an object that is modifiable as a sequence of immutable *versions*; each version is the state of the object after an update is made to the object. Synchronization can then be treated as a mechanism for naming versions to be read and for defining where in the sequence of versions the version resulting from some update should be placed. In systems based on mutual exclusion, the timing of accesses selects the versions accessed. In the system developed here, called NAMOS, versions have two component names consisting of the name of an object and a pseudo-time, the name of the system state to which the version belongs. By giving programs control over the pseudo-time in which an access is made, synchronization of accesses to multiple objects is simplified.

Namos is intended to be used in an environment where unreliable components, such as communication lines and processors, and autonomous control of resources occasionally cause certain objects to become inaccessible, perhaps in the middle of an atomic transaction. Computations may also suddenly halt (perhaps as the result of a system crash) never to be restarted. NAMOS provides facilities for recovering from such sudden failures, grouping updates into sets called possibilities, such that failure of any update belonging to a possibility prevents all of the other updates in the possibility. The naming mechanism of NAMOS also provides a useful tool for restoring a consistent state of the system after a failure resulting in irrecoverable loss of information or a user mistake resulting in an inconsistent state.

An important motivation for the development of NAMOS is the need to support decentralized development of application systems by combining existing applications systems that deal with shared data. NAMOS supports the construction of modules that locally ensure their own correct synchronization and recovery from inaccessibility. Larger modules that use several separately designed modules can then be constructed, perhaps with additional synchronization constraints, without modifying the modules used. In most systems based on mutual exclusion, such *post hoc* integration of modules is difficult or impossible. {AD A061-407}

TR-206
Borkin, S.A.
EQUIVALENCE PROPERTIES OF SEMANTIC DATA MODELS FOR DATA BASE SYSTEMS
Pages: 311 Ph.D dissertation/January 1979 $12.65
Keywords: data base systems, data models, data model equivalence, semantic relation data model, semantic graph data model
Abstract: A *data model* defines the types of structures present in a data base and the types of operations which may be used to alter the data base. An understanding of *data model equivalence properties* is necessary if one wishes to implement a system which presents different users with views of a data base in terms of differing data models or which provides a common interface to several data base systems defined in terms of different data models. Requisites for this understanding are formal definitions of the involved concepts and a formal framework in which different data models can be compared.

This thesis presents formal definitions of the terms *data base*, *operation*, *operation type*, *application model* and *data model*. *Data base state equivalence*, *operation equivalence*, *application model equivalence* and *data model equivalence* are distinguished. Three types of application and data model equivalence are defined - *isomorphic*, *composed operation* and *state dependent*. Implementation implications of these different equivalences are discussed.

This definitional framework is used to explore the equivalence properties of two *semantic data models*. Semantic data models are data models whose structures are meant to have clear interpretations in terms of the applications being modeled. The *semantic graph* and *semantic relation data models* are formally defined. The definition of the semantic relation data model includes the definitions of *constraints* and the *semantic relational algebra*.

It is proved that the semantic graph and the *restricted* semantic relation data models are state dependent equivalent. Observations on the network vs. relational data model "controversy" are presented. Suggestions for applications of this research include a *dual semantic data model interface.* {AD A066-386}

TR-207
Montgomery, W.A.
ROBUST CONCURRENCY CONTROL FOR A DISTRIBUTED INFORMATION SYSTEM
Pages: 191 Ph.D. Dissertation/January 1979 $9.05
Keywords: concurrency control, distributed systems, message passing
Abstract: This dissertation presents a collection of protocols for coordinating transactions in a distributed information system. The system is modeled as a collection of processes that communicate only through message passing. Each process manages some portion of the data base, and several processes may cooperate in performing a single transaction.

The thesis presents a model for computation in a distributed information system in which the sites and communication links may fail. The effects of such failures on the computation are described in the model. The thesis discusses implementation techniques that could be used to limit the effects of failures in a real system to those described in the model.

A hierarchical protocol for coordinating transactions is presented. The accesses to be performed during a transaction are pre-analyzed to select the protocols needed to coordinate the processes that participate in the implementation of the transaction. This analysis can be used to guide the organization of the data base so as to minimize the amount of locking required in performing frequent or important transactions. An important aspect of this mechanism is that it allows transactions that cannot accurately be pre-analyzed to be performed and correctly synchronized without severely degrading the performance of the system in performing more predictable transactions.

A novel approach to the problem of making updates at several different sites atomically is also discussed. This approach is based on the notion of a polyvalue, which is used to represent two or more possible values for a single data item. A polyvalue is created for an item involved in an update that has been delayed due to a failure. By assigning a polyvalue to such an item, that item can be made accessible to subsequent transactions, rather than remaining locked until the update can be completed. A polyvalue describes the possible values that may be correct for an item, depending on the outcome of transactions that have been interrupted by failures. Frequently, the most important effects of a transaction (such as the payment of money) can be determined without knowing the exact values of the items in the data base. A polyvalue for an item that is accessed by such a transaction may be sufficient to determine such effects. By using polyvalues, we can guarantee that a data item will not be made inaccessible to any failure other than a failure of the site that holds the item.

A strong motivation for the development of these protocols is the desire that the individual sites of a distributed information system fail independently, and that a site or a group of sites be able to continue local processing operations when a failure has isolated them from the rest of the sites. Many of the previous coordination mechanism have only considered the continued operation of the sites that remain with the system to be important. Another motivating factor for the development of these protocols is the idea that in many applications, the processing to be performed exhibits a high degree of locality of reference, in that most operations involve only a small number of sites. By structuring the coordination mechanism to take advantage of this locality of reference, one can have protocols that are simple, efficient, and robust for the particular application. {AD A066-996}

TR-208
Krizan, B. C.
A MINICOMPUTER NETWORK SIMULATION SYSTEM
Pages: 123 S.B. & S.M. Thesis/February 1979 $7.00
Keywords: computer networks, networks
Abstract: The design, development and use of cost-effective computer networks require information about system behavior given a variety of network structures and operational policies. Because computer networks are complex systems whose behavior is generally not intuitively understood, there is a need for system analysis tools to provide a wide range of performance information.

This thesis describes a simulation system that generates behavioral information for a class of minicomputer network systems. This simulation system is modularly designed with modules for network modeling, specification of the network processing load, and simulation (a discrete event simulator). The network modeling done with the simulation system is based on a general purpose discrete modeling discipline. Flexible network model building blocks made from the basic modeling discipline structures are provided to the simulation system user. To prepare a simulation experiment the user assembles a network model from the building blocks and specifies a network processing load. To generate performance information the network model and load specification are input to the simulator along with simulation control parameters. On completion of the simulation experiment the generated performance information is output in a palatable form to the user. Overall this simulation system is a convenient and flexible system analysis tool for minicomputer networks.

TR-209
Snyder, A.
A MACHINE ARCHITECTURE TO SUPPORT AN OBJECT-ORIENTED LANGUAGE
Pages: 182 Ph.D. Dissertation/March 1979 $8.80
Keywords: language-oriented machine architecture, object oriented programming, garbage collection, reference counts
Abstract: In object-oriented languages (e.g., LISP, Simula, and CLU), all (or most) data objects used by a program are implicitly allocated from a free-storage area and are accessed via fixed-size references. The storage for an object is automatically reclaimed (garbage collected) when the object is no longer accessible to the program.

This thesis presents the design of a computer system that directly supports an object-oriented machine language. The machine provides a single, large universe of objects shared by multiple processes. The design uses expected future technologies (fast-access secondary storage devices and inexpensive processors) to satisfy the goals of good performance and a simple, modular system organization.

Automatic storage reclamation is performed primarily using reference counts. The proposed reference count implementation reduces the time overhead of automatic storage reclamation and allows most reclamation processing to be performed in parallel with normal computation. In addition, the reference count scheme can be used in a multiprocessor configuration without introducing complex synchronization problems.
idual transactions. Using a simple transaction model we show that recognizing the transaction histories which are serializable is an NP-complete problem. We therefore introduce several efficiently recognizable subclasses of the class of serializable histories; most of these subclasses correspond to serializability principles existing in the literature and used in practice. We also propose two new principles which subsume all previously known ones. We give necessary and sufficient conditions for a class of histories to be the output of an efficient history scheduler; these conditions imply that there can be no efficient scheduler that outputs all of serializable histories studied above have an efficient scheduler. Finally, we show how our results can be extended to far more general transaction models, to transactions with partly interpreted functions, and to distributed data base systems.

TR-210
Papadimitriou, C.H.
SERIALIZABILITY OF CONCURRENT DATA BASE UPDATES
Pages: 56 March 1979 $5.00
Keywords: concurrent data base, serializability, NP-complete
Abstract: A sequence of interleaved user transactions in a data base system may not be *serializable,* i.e., equivalent to some sequential execution of the individual transactions. Using a simple transaction model we show that recognizing the transaction histories which are serializable is an NP-complete problem. We therefore introduce several efficiently recognizable subclasses of the class of serializable histories; most of these subclasses correspond to serializability principles existing in the literature and used in practice. We also propose two new principles which subsume all previously known ones. We give necessary and sufficient conditions for a class of histories to be the output of an efficient history scheduler, these conditions imply that there can be no efficient scheduler that outputs all of serializable histories studies above have an efficient scheduler. Finally, we show how our results can be extended to far more general transaction models, to transactions with partly interpreted functions, and to distributed data base systems.

TR-211
Bloom, T.
SYNCHRONIZATION MECHANISMS FOR MODULAR PROGRAMMING LANGUAGES
Pages: 116 S.M. Thesis/April 1979 $6.80
Keywords: synchronization, concurrency, modularity, data abstractions, programming methodology
Abstract: Any programming language that supports concurrency needs a synchronization construct with which to express access control for shared resources. This thesis examines synchronization constructs from the standpoint of language design for reliable software. The criteria a synchronization mechanism must satisfy to support construction of reliable, easily maintainable concurrent software are defined. Some of these criteria, such as expressive power, can be defined only with respect to the set of problems the mechanism is expected to handle. A definition of the range of problems considered to be synchronization problems is therefore needed. Such a definition is provided by describing the possible types of constraints that may be imposed on access to shared resources. We then use this taxonomy of synchronization constraints to develop techniques for evaluating how well synchronization constructs meet the criteria discussed. These techniques are then applied to three existing synchronization mechanisms: monitors, path expressions, and serializes. Evaluations are presented, and the three mechanisms compared. {AD A069-819}

TR-212
Rabin, M.O.
DIGITALIZED SIGNATURES AND PUBLIC-KEY FUNCTIONS AS INTRACTABLE AS FACTORIZATION
Pages: 16 January 1979 $3.80
Keywords: public-key cryptosystems, digitalized signatures, factorization, intractable problems, cryptography, security
Abstract: We introduce a new class of public-key functions involving a number n = pq having two large prime factors. As usual, the key n is public, while p and q are the private key used by the issuer for production of signatures and function inversion. These functions can be used for all the applications involving public-key functions proposed by Diffie and Hellman, including digitalized signatures. We prove that for any given n, if we can invert the function $y = E_n(x1)$ for even a small percentage of the values y then we can factor n. Thus as long as factorization of large numbers remains practically intractable, for appropriately chosen keys not even a small percentage of signatures are forgable. Breaking the RSA function is at most hard as factorization, but is not known to be equivalent to factorization even in the weak sense that ability to invert all function values entails ability to factor the key. Computation time for these functions, i.e. signature verification, is several hundred times faster than for the RSA scheme. Inversion time, using the private key, is comparable. The almost-everywhere intractability of signature-forgery for our functions (on the assumption that factoring is intractable) is of great practical significance and seems to be the first proved result of this kind.

TR-213
Rabin, M.O.
PROBABILISTIC ALGORITHMS IN FINITE FIELDS
Pages: 26 January 1979 $4.10
Keywords: computations in finite fields, root finding, factorization, probabilistic algorithms
Abstract: We present probabilistic algorithms for the problems of finding an irreducible polynomial of degree *n* over a finite field, finding roots of a polynomial, and factoring the polynomial into its irreducible factors over a finite field. All of these problems are of importance in algebraic coding theory, algebraic symbol manipulation, and number theory. These algorithms have a very transparent, easy to program structure. For finite fields of large characteristic *p*, so that exhaustive search throng z_p is not feasible, our algorithms are of lower order in the degrees of the polynomial and fields in question than previously published algorithms.

TR-214
McLeod, D.
A SEMANTIC DATA BASE MODEL AND ITS ASSOCIATED STRUCTURED USER INTERFACE
Pages: 378 Ph.D. Dissertation/March 1979 $14.65
Keywords: data base management, data models, data semantics, data base modeling, data definition, data base design
Abstract: The conventional approaches to the structuring of data bases provided in contemporary data base management systems are in many ways unsatisfactory for modeling data base application environments. The features they provide are too low-level, computer-oriented, and representational to allow the semantics of a data base to be directly expressed in its structure. The *semantic data model* **SDM** has been designed as a natural application modeling mechanism that can capture and express the structure of an application environment. The features of the SDM correspond to the principal intentional structures naturally occurring in contemporary data base applications. The SDM provides a rich but limited vocabulary of data structure types and primitive operations, striking a balance between semantic expressibility and the control of complexity. Furthermore, facilities for expressing derived (conceptually redundant) information are an essential part of the SDM; derived information is as prominent in the description of an SDM data base as is primitive data.

The DSM is designed to enhance the effectiveness and usability of data base systems:

1. SDM data bases are to a large extent self-documenting, in the sense that the description and structure of a data base are expressed in terms which are close to those used by users in describing the application environment.

2. The DSM can support powerful user interface facilities, and can improve the user interface effectiveness for a variety of types of users (with varying needs and abilities). Significantly, SDM data bases capture information in a form useful to its users, and allow derived information helpful in new data base uses to be defined in the data base structure. In particular, the SDM supports an incremental, interactive interface for the "naive" nonprogrammer *an interaction formulation advisor*, which guides the user through the data base and the process of formulating a query or update request.

3. The SDM can be used as a tool in the data base design process. The SDM aids in the identification of relevant information in a data base application environment, as well as in organizing that information and relating it to its possible uses. This can grealty improve the design of lower-level, conventional data bases.

The use of the SDM is not dependent on the successful implementation of a new data base management system that directly supports it. There are many data base management systems in use today which represent a considerable investment on the part of their developers and users; the SDM can be effectively used in conjunction with these existing data base systems to enhance their effectiveness and usability. For example, a prototype interaction formulation advisor demonstrates that the SDM can be used as a user-oriented 'front end" to a conventional data base system; an analysis of application modeling with the SDM illustrates its effectiveness in improving and simplifying the data base design process. {AD A068-112}

TR-215
Svobodova, L., Liskov, B., Clark, D.D.
DISTRIBUTED COMPUTER SYSTEMS: STRUCTURE AND SEMANTICS
Pages: 58 April 1979 $5.05
Keywords: distributed computer systems, programming languages, operating systems
Abstract: This report describes an ongoing project in the area of design of distributed systems. The goal is to develop an effective programming system that will support well-structured design implementation, maintenance and control of distributed processing applications. This programming system combines a powerful high level language and operating system features, and addresses the underlying system problems that affect the reliability and security perceived on the application level. The report presents a conceptual model of distributed computation, and, in the context of this model, discusses our approaches to inter-node communication and cooperation, reliability, and protection. One of the basic goals of our project is to allow the application programmer to work with applicatin-oriented entities. Thus, inter-node messages, error handling and protection constraints will all be expressible in application oriented terms. The report concludes with some examples of the language constructs and an outline of the future research under this project. {AD A070-286}

TR-216
Myers, J. M.
ANALYSIS OF THE SIMPLE CODE FOR DATA FLOW COMPUTATION
Pages: 122 May 1979 $7.00
Keywords: data flow, algorithm analysis, parallel computation, computational hydrodynamics, role diagram
Abstract: We analyze a problem in hydrodynamics from the standpoint of computation on a data flow computer that is not yet fully specified, with the objectives of helping to further specify the computer and helping to develop VAL as its source language. Lawrence Livermore Laboratory supplied the algorithm for hydrodynamics, including heat flow, as a 1749-line FORTRAN code called SIMPLE.

The algorithm viewed as 'abstract' (i.e. independent of physical arrangements in space and time for its realization) is shown to imply spatial and temporal structure that must appear in any and all implementations. Both for hardware design and program compilation it is useful to map this structure to grosser levels of description, with the grosser levels reflecting modularity of computational resources conjoined with modularity of the algorithm. Following Holt (1979) we use role diagrams to display spatio-temporal structure at different descriptive levels, so as to guide translation into VAL as well as the analysis of the time to compute.

Inter-resource communication essential to the problem is displayed, and various issues of machine design are defined. Using VAL with one set of extensions, we express the algorithm so that in principle it can be compiled for execution by a data flow computer. Input-output functions beyond those implied by the SIMPLE code are discussed. A second set of extensions to VAL is advocated to express the conjunction of problem and resource modularity, so as to guide compilation. The dependence of time to compute on the number of processing units is shown for various aspects of the problem.

TR-217
Brown, D.
STORAGE AND ACCESS COSTS FOR IMPLEMENTATIONS OF VARIABLE LENGTH LISTS
Pages: 208 Ph.D. Dissertation/April 1979 $9.55
Keywords: storage, access
Abstract: Consider a machine with a cellular memory used to store a list X^i, where X is a finite alphabet and $i \in N$. We investigate the machine representation of such a list and the implementation of common list operations such as determining the i^{th} element and adding or deleting an element. Information-theoretic arguments are used in order to obtain lower bounds on storage and access costs for implementing variable-length lists and, in particular, stacks. Representations are discussed which attain these bounds separately and can sometimes attain both, although it is shown that some common representations of stacks cannot simultaneously achieve both. On the constructive side, we show that it is possible to implement a stack of any finite length so as to achieve Kraft storage and so that the number of memory cell accesses required to perform a PUSH or a TOP operation is always O(log n) but where, assuming a non-increasing probability distribution on stack lengths, a POP operation requires on the average only a constant number of accesses.

TR-218
Ackerman, W.B., Dennis, J.B.
VAL--A VALUE-ORIENTED ALGORITHMIC LANGUAGE, PRELIMINARY REFERENCE MANUAL
Pages: 80 June 1979 $5.75
Keywords: programming languages, applicative languages, VAL
Abstract: The programming language **VAL** (*Value-Oriented Algorithmic Language*) is designed for expressing algorithms for execution on computers capable of highly concurrent operation. More specifically, the application area to be supported is numerical computation which strains the limits of high performance machines, and the primary targets for translation of VAL programs are data driven machines of the form under development by the Computation Structures Group of the MIT Laboratory for Computer Science for high performance numerical computation. {AD A072-394}

TR-219
Sollins, K.R.
COPYING COMPLEX STRUCTURES IN A DISTRIBUTED SYSTEM
Pages: 109 S.M. Thesis/July 1979 $6.60
Keywords: sharing, distributed systems, message passing, strongly typed objects
Abstract: This thesis presents a model of a distributed system. the universe of objects in the distributed system is divided into mutually exclusive sets, each set corresponding to a context. This model allows naming beyond the context boundaries, but limits communications across such boundaries to message passing only. Copying of complex data structures is investigated in this model, and semantics, algorithms, and sample implementations are presented for three candidate copy operations. Of particular interest is a new *copy-full-local* which copies a complex data structure to the boundaries of the context containing the object. {AD A072-441}

TR-220
Kosinski, P.R.
DENOTATIONAL SEMANTICS OF DETERMINATE AND NON-DETERMINATE DATA FLOW PROGRAMS
Pages: 104 Ph.D. Dissertation/July 1979 $6.45
Keywords: semantics, data flow, denotational semantics, non-determinate programs, parallel computation
Abstract: Among its other characteristics, a programming language should be conducive to writing modular program's, be able to express parallelism and non-determinate behavior, and it should have a cleanly formalizeable semantics. Data flow programming languages have all these characteristics and are especially amenable to mathematization of their semantics in the denotational style of Scott and Strachey. Many real world programming problems, such as operating systems and data base inquiry systems, require a programming language capable of non-determinacy because of the non-determid the relation of this approach to other approaches. In particular, it is unnecessary to use the "power domain" construction in order to handle simple non-determinacy in data flow languages.

TR-221
Berzins, V.A.
ABSTRACT MODEL SPECIFICATIONS FOR DATA ABSTRACTIONS
Pages: 175 Ph.D. Dissertation/July 1979 $8.60
Keywords: specification, abstract data types, data abstractions, data types, errors, side effects, modification of shared data, verification
Abstract: A data abstraction introduces a data type with a hidden representation. Specifications of data abstractions are required to allow the data to be described and used without reference to the underlying representation. There are two main approaches to specifying data abstractions, the abstract model approach and the axiomatic approach.

This thesis is concerned with the problems of formalizing and extending the abstract model approach. A formally defined language for writing abstract model specifications is presented. the correctness of an implementation with respect to an abstract model specification is defined, and a method for proving the correctness of implementations is proposed.

Our formulation treats data abstractions with operations that can dynamically create new data objects, modify the properties of existing data objects, and raise exception conditions when presented with unusual input values.

TR-222
Halstead, R.H.
REFERENCE TREE NETWORKS: VIRTUAL MACHINE AND IMPLEMENTATION
Pages: 252 Ph.D. Dissertation/September 1979 $10.90
Keywords: message passing, distributed computing, multiprocessor systems, distributed object management, networks
Abstract: A current-technology computing machine may be roughly decomposed into a processor, a memory, and a data path connecting them. The interposition of this data bath between processing and storage elements creates a bottleneck, which inhibits progress at the high-performance end of the technological spectrum. Additionally, the monolithic nature of present-day processors resists incremental addition or removal of processing power.

The research described here attacks the problem of constructing more powerful and more flexible computer systems along three fronts: the definition of a virtual machine providing for parallel computation using objects and object references, the development of a distributed implementation mechanism ("reference trees") supporting object management functions including garbage collection, and the reinvestigation of scheduling algorithms and collection performance results.

A *reference tree network* using these concepts is composed of a multitude of independent small processors, yet operates as a coherent programming system. Programs and data spread automatically and transparently through the network to occupy underused resources. The modular structure of the network provides many parallel data paths as well as allowing for easy addition or removal of modules, thus addressing some of the problems discussed above. A prototype reference tree network, the MuNet, is currently in operation. {AD A076-570}

TR-223
Brown, G.P.
TOWARD A COMPUTATIONAL THEORY OF INDIRECT SPEECH ACTS
Pages: 207 October 1979 $9.55
Keywords: indirect speech acts, speech acts, english dialog, conversational maxims, semantics, natural language, computational linguistics, pragmatics,language recognition
Abstract: The variety of surface forms that may be used to convey a given speech act pose a major problem in modeling task-oriented (and other) dialogues. Many such forms are so-called indirect speech acts, that is, surface form does not correspond to the (or one) intended speech act. While this topic has received extensive attention from linguists, their concerns have not usually been computationally motivated. In this paper, I present a non computational analysis of indirect speech act forms with an eye to computational considerations.

The paper is divided into two parts. Part 1 presents categories and rules for indirect speech acts, justified where possible by traditional linguistic arguments. The second part of the paper draws a set of computational implications from the material presented in Part 1. This is done within the general framework of a process model of recognition. Part 2 contains a discussion of the basic types of mechanisms needed for the classes of indirect speech act identified in Part 1. The discussion includes an examination of the dependencies between processes and an initial categorization of the types of knowledge that must be considered in interpreting indirect speech acts. {AD A077-065}

TR-224
Isaman, D.L.
DATA-STRUCTURING OPERATIONS IN CONCURRENT COMPUTATIONS
Pages: 563 Ph.D. Dissertation/October 1979 $20.20
Keywords: data flow architecture, determinacy, models of concurrent computation, structure memory
Abstract: This thesis proposes operational specifications for a Structure Memory (SM). A specialized hardware component of a general-purpose computing system, the SM would directly execute operations on dynamically structured data stored in it. The computing system is assumed capable of exploiting program concurrency at the machine-instruction level. For explanatory purposes, the proposed structure operations are presented in the context of the data flow model of concurrent computation.

Concurrency among a set of program instructions which all examine or modify the same structure must be carefully controlled, if the program is to be determinate. The first of two major contributions of the thesis is a combination hardware/software discipline which affords maximal concurrency consistent with determinacy. Its key feature is that the SM will not return a given pointer until certain previously-returned pointers to the same structure are no longer available as operands.

The second major contribution in the entry-execution model of concurrent computation. Reversing the emphasis of most previous work, this model concentrates on the operations performed by instructions, while abstracting away details of how operands are passed among them and how their execution order is determined. The essence of structure operators - that the result of an execution of one of may depend on the input to previous executions of that and other operators - is given a natural expression in the new model. A proof of sufficient conditions for determinacy of a program containing structure operators is made more generally applicable through use of the entry-execution model as its medium.

TR-225
Liskov, B., Atkinson, R.R., Bloom, T., Moss, E.B., Schaffert, C., Scheifler, R., Snyder, A.
CLU REFERENCE MANUAL
Pages: 166 October 1979 $8.30
Keywords: programming languages, data abstractions, strong type checking, modularity, exception handling, iteration abstractions, CLU
Abstract: This document serves both as an introduction to CLU and as a language reference manual. Sections 1 through 4 present an overview of the language. These sections highlight the essential features of CLU, and discuss how CLU differs from other, more conventional, languages. Sections 5 through 13 form the reference manual proper. These sections describve each aspect of CLU in detail, and discuss the proper use of various features. Appendices 1 through III provide concise summaries of CLU's syntax, data types, and I/O facilities. Appendix IV contains example programs. {AD A077-018}

TR-226
Reuveni, A.
THE EVENT BASED LANGUAGE AND ITS MULTIPLE PROCESSOR IMPLEMENTATIONS
Pages: 288 Ph.D. Dissertation/January 1980 $11.95
Keywords: programming languages, parallel programming, concurrency, synchronization primitives, events, ,event handlers, EBL, modularity, networks, dataflow, NP-complete
Abstract: This research defines and analyzes a simple language for parallel programming which is designed for multiple processor systems. The language (EBL) is based on events rich provide the only control mechanism. Events are explicitly caused by the program, and they activate instances of dynamic program units called event handlers. The only operation that can be performed by an instance of an event handler is the causing of new events. The language constructs are primitive; never-the-less, the capability of hierarchical program design is provided via static modules and other modularity sources.

The language does not contain conventional constructs such as: variables, assignment statements, goto statements, iteration constructs, procedures, functions, and semaphores; however, these can be easily modeled. In addition, events allow activation of parallel processes, synchronization of parallel processes, mutual exclusion, message passing, immutable objects, and the effect of mutable objects.

Schemes for implementation of the language on processor networks are investigated. An implementation scheme based on communicating managers which operate without any centralized control is described. A relaxed distributed locking algorithm in which deadlocks are prevented is developed; it does not assume a total order on all objects to be locked. Several optimization problems, e.g., optimal distribution of objects in a network, are investigated. The problems are shown to be NP-hard and heuristic algorithms are suggested. Implementation schemes of the language on a data flow processor are described. These add to the processor the capability of procedures, and synchronization primitives such as semaphores. {AD A081-950}

TR-227
Rosenberg, R.L.
INCOMPREHENSIBLE COMPUTER SYSTEMS: KNOWLEDGE WITHOUT WISDOM
Pages: 117 S.M. Thesis/January 1980 $6.85
Keywords: incomprehensible systems, computer systems, technological incomprehensibility, technological systems, societal implications of computers
Abstract: An analysis of the incomprehensibility of large, complex computer systems is made. The thesis is that there is strong relationship between system incomprehensibility and the necessity to trust computer systems. A cogent definition of incomprehensibility in computer systems is established, with common themes drawn from interdisciplinary literature dealing with computers and society. Reasons for the creation of incomprehensible computer systems are explored, as well as the consequences (both technical and social) of using and relying on them. The relationship between the real and perceived purposes of computer systems and the appropriateness of trusting these systems is analyzed. Approaches for dealing with the existence of vital computer systems which are functionally incomprehensible are evaluated, and positive suggestions are made.

TR-228
Weng, K.-S.
AN ABSTRACT IMPLEMENTATION FOR A GENERALIZED DATA FLOW LANGUAGE
Pages: 154 Ph.D. Dissertation/January 1980 $7.95
Keywords: computer organization, concurrency, programming languages, data flow architecture
Abstract: The expressiveness of a programming language strongly affects how a computer architecture can attain high performance through concurrent operation of hardware. In this thesis, we demonstrate that a suitable computation model can provide a basis both for a good programming language and for an architecture that fully exploit the inherent concurrency in algorithms expressed in the language. To thiscy that merges two sequences of values in a nondeterminate manner.

The main advantage of architectures based on data flow concepts is that concurrency at the level of primitive operations can easily be exploited. The architecture presented in this thesis is based on a form of processor proposed by Dennis and Misunas. We give an effective implementation of the language on this architecture so concurrency in procedure activations and operations on streams and data structures can be exploited. Novel architectural concepts are suggested for addressing coherence and contention problems in supporting concurrent accesses to data structures.

TR-229
Atkinson, R.R.
AUTOMATIC VERIFICATION OF SERIALIZES
Pages: 203 Ph.D. Dissertation/March 1980 $9.40
Keywords: verification, concurrency, monitors, serializers, specification
Abstract: This thesis is concerned with the problem of controlling concurrent access to shared data. A language construct is proposed to enforce such control; a specification language is defined to describe the formal requirements of such control; and verification techniques are given to prove that instances of the construct satisfy their specifications. The techniques are justified in terms of the definition of the construct and the definition of the specification language. Results are given for a program that implements a number of the techniques, illustrated by verifying several versions of the readers-writers problem. Interactions between instances of the construct are discussed in the context of a simple file system. {AD A082-885}

TR-230
Baratz, A.E.
THE COMPLEXITY OF THE MAXIMUM NETWORK FLOW PROBLEM
Pages: 68 S.M. Thesis/March 1980 $5.35
Keywords: computational complexity, geometric complexity, network flow, polyhedral decision problem

Abstract: This thesis deals with the computational complexity of the maximum network flow problem. We first introduce the basic concepts and fundamental theorems upon which the study of "max-flow" has been built. We then trace the development of max-flow algorithms from the original "labeling algorithm" of Ford and Fulkerson, through a recent $0(V^*E^*\log^2V)$ algorithm due to Galil and Naamad. We include a description of each of these algorithms, alone with a proof of correctness and proof of running time for most of them. Finally we turn our attention to the problem of establishing lower bounds on the complexity of max-flow. We show that a straightforward application of the polyhedral lower bound technique developed by Yao, Avis and Rivest fails to produce a non-linear lower bound on max-flow. In the process, however, we prove several interesting results concerning the facial structure of a class of polyhedra very closely related to the maximum network flow problem.

TR-231
Jaffe, J.M.
PARALLEL COMPUTATION: SYNCHRONIZATION, SCHEDULING, AND SCHEMES
Pages: 264 Ph.D. Dissertation/March 1980 $11.25
Keywords: parallel computation, synchronization, scheduling, schemes
Abstract: There are two primary means of resource allocation in computer systems. There is the powerful mechanism of using a centralized resource manager that allocated the resources. An apparently weaker mechanism is for the asynchronous processes of the system to allocate resources with some type of message passing between themselves. This thesis provides a unifying treatment of these two methods. It is shown that a managed system may be simulated by the processes. As a corollary, a wide variety of synchronization algorithms may be accomplished without a manager.

The simulation works correctly even in an environment with unreliabilities. Processes may die in an undetectable manner and the memory of the system may be faulty. Thus our general simulation provides the first known algorithm for synchronizing dying processes in a faulty memory environment.

Scheduling jobs on processors of different capabilities is studied. Algorithms are presented for two machine environments that have better performance than previously studied algorithms. These environments are processors of uniformly different speeds with partially ordered tasks and unrelated processors with independent tasks. In addition, the class of all preemptive schedules for uniform processor systems are studied. These schedules are shown to be more effective than previous analyses.

Scheduling jobs on functionally dedicated processors is introduced. Algorithms are presented for scheduling jobs on such processors, both in the case that the processors are equally fast and in the case that they are of different speeds.

The expressive power of the data flow schemas of Dennis is evaluated. It is shown that data flow schemes have the power to express an arbitrary determinated functional. The proof involves a demonstration that "restricted data flow schemes" can simulate Turing Machines. This provides a new, simple basis for computability.

TR-232
Luniewski, A.
THE ARCHITECTURE OF AN OBJECT BASED PERSONAL COMPUTER
Pages: 265 Ph.D. Dissertation/March 1980 $11.30
Keywords: personal computers, capability addressing, structured programming, operating systems, high level machine architecture
Abstract: This thesis proposes the architecture of a personal computer that provides better support that conventional architectures for recently developed concepts of structured programming. The architecture separates implementation and high level language issues. The architecture eliminates the need for an operating system by including, in a language independent manner, the features normally found in operating systems. The architecture allows multiple languages to coexist safely. It is complete; the user has no need to leave the world defined by the architecture to solve a problem, including the important case of executing entrusted programs.

The architecture provides the semantic base needed by most languages. It supports a flexible execution environment that treats executable code and naming environments as objects. It explicitly supports the termination model of exception handling. A new mechanism, object viewers, provides type extension, access restriction and access revocation. The operating system features of process, inter-process communication/synchronization, permanent storage, I/O and system initialization/shutdown are provided.

An efficient implementation of the architecture is presented that is suitable for a personal computer. The implementation provides a large, real-time garbage collected object heap built out of a physical multi-level memory system.

Some ways in which the architecture can be used are shown. The focus is on showing how some problems of language implementation can be handled. {AD A083-433}

TR-233
Kaiser, G.E.
AUTOMATIC EXTENSION OF AN AUGMENTED TRANSITION NETWORK GRAMMAR FOR MORSE CODE CONVERSATIONS
Pages: 97 S.B. Thesis/April 1980 $6.25
Keywords: morse code, natural language
Abstract: This report describes a 'learning program' that acquires much of the knowledge required by a parsing system that processes conversations in a 'natural' language akin to ham-radio jargon. The learning program derives information from example sentences taken from transcripts of actual conversations, and uses this knowledge to extend the 'core' augmented transition network, (ATN), grammar. The parser can use the extended grammar to process the example sentences, plus a large number of syntactically and semantically related sentences.

The learning program uses a set of heuristics to determine the difference between the existing version of the grammar and a superset that could process the example sentence. A set of models act as templates to produce possible extensions and adds it to the grammar. This extension is henceforth an integral component of the knowledge base and may be used by the parser to process conversations and by the learning program to extend the grammar further.

This report relates the mechanisms used by the learning program to grammatical inference of context-sensitive languages, which include the natural languages, and some proposed linguistic models of human language acquisition. These models describe language acquisition as a process of developing hypotheses according to the constraints of innate universal rules, and acceptance of those hypotheses that make it possible for the child to understand new sentences. Similarly, the learning program develops its hypotheses within the constraints of certain 'universal' models and accepts only those hypotheses that enable the parser to process the motivating example. {AD A084-411}

TR-234
Herlihy, M.P.
TRANSMITTING ABSTRACT VALUES IN MESSAGES
Pages: 121 S.M. Thesis/May 1980 $6.95
Keywords. abstract data types, distributed systems, message passing, modularity, object oriented programming, programming languages, programming methodology, CLU
Abstract: This thesis develops primitives for a programming language intended for use in a distributed computer system where individual nodes may have different hardware or software configurations. Our primitives are presented as extensions to the CLU language. We assume that differences in hardware and in administrative policy require that individual nodes be free to choose their own local representations for common types, including user-defined types. Our main objective is to provide primitives to communicate values of user-defined type. Our primitives support a large degree of node autonomy, without requiring that communicating nodes have prior knowledge of one another's special characteristics. We argue that the precise meaning of value transmission is type-dependent; thus the user, not the language, must control the meaning of transmission for values of a type. {AD A086-984}

TR-235
Levin, L.A.
A CONCEPT OF INDEPENDENCE WITH APPLICATIONS IN VARIOUS FIELDS OF MATHEMATICS
Pages: 21 May 1980 $3.95
Keywords: complexity, randomness, independence
Abstract: We use Kolmogorov's algorithmic approach to information theory to define a concept of independence of sequences, or equivalently, the boundedness of their mutual information. This concept is applied to probability theory, intuitionistic logic, and the theory of algorithms. For each case, we study the advantage of accepting the postulate that the objects studied by the theory are independent of any sequence determined by a mathematical property.

TR-236
Lloyd, E.L.
SCHEDULING TASK SYSTEMS WITH RESOURCES
Pages: 146 Ph.D. Dissertation/May 1980 $7.70
Keywords: task system with resources, concurrent systems, resource constrained scheduling
Abstract: Minimum execution time scheduling of task systems with resources has been the subject of several papers over the past few years. The model used for much of this work assumes that the resources associated with computer systems - readers, printers, disk drives - are not "continuous" resources. We present an alternative model of task systems with resources in which the resources are discrete. That is, there are a specific number of indivisible units of each resource and a task may require only integral numbers of those units. Several results involving the worst case performance of list scheduling and critical path scheduling with respect to this model are given. A new result on critical path scheduling of task systems with continuous resources is also given. Finally, a comparison will be made between corresponding bounds for the continuous and discrete models.

TR-237
Kapur, D.
TOWARDS A THEORY FOR ABSTRACT DATA TYPES
Pages: 251 Ph.D. Dissertation/June 1980 $10.85
Keywords: abstract data types, data types, theory of computation, programming languages
Abstract: A rigorous framework for studying immutable data types having nondeterministic operations and operations exhibiting exceptional behavior is developed. The framework embodies the view of a data type taken in programming languages, and supports hierarchical and modular structure among data types.

The central notion in this framework is the definition of a data type. An algebraic and behavioral approach for defining a data type is developed which focuses on the input-output behavior of a data type as observed through its operations. The definition of a data type abstracts from the representational structure of its values as well as from the multiple representations of the values for any representational structure.

A hierarchical specification language for data types is proposed. The semantics of a specification is a set of related data types whose operations have the behavior captured by the specification. A clear distinction is made between a data type and its specification(s). The normal behavior and the exceptional behavior of the operations are specified separately. The specification language provides mechanisms to specify (i) a precondition for an operation thus stating its intended inputs, (ii) the exceptions which must be signalled by the operations, and (iii) the exceptions which the operations can optionally signal. Two properties of a specification, consistency and behavioral completeness, are defined. A consistent specification is guaranteed to specify at least one data type. A behaviorally complete specification 'completely' specifies the observable behavior of the operations on their intended inputs.

A deductive system based on first order multi-sorted predicate calculus with identity is developed for abstract data types. It embodies the general properties of data types, which are not explicitly stated in a specification. The theory of a data type, which consists of a subset of the first order properties of the data type, is constructed from its specification. The theory is used in verifying programs and designs expressed using the data type. Two properties of a specification, well definedness and completeness, are defined based on what can be proved from it using different fragments of the deductive system. The sufficient completeness property of Guttag and Horning is also formalized and related to the behavioral completeness property. The well definedness property is stronger than the consistency property, because the well definedness property not only requires that the specification specifies at least one data type, but also captures the intuition that it preserves other specifications used in it thus ensuring modular structure among specifications. The completeness property is stronger than the sufficient completeness property, since in addition to the requirement that the behavior of the observers can be deduced on any intended input by equational reasoning, it also requires that the equivalence of the observable effect of the constructors can be deduced from the specification by equational reasoning.

A correctness criterion is proposed for an implementation coded in a programming language with respect to a specification. It is defined as a relation between the semantics of an implementation and the semantics of a specification. It does not require a correct implementation to have the maximum amount of nondeterminism specified by a specification. A methodology for proving correctness of an implementation is developed which embodies the correctness criterion. {AD A085-877}

TR-238
Bloniarz, P.A.
THE COMPLEXITY OF MONOTONE BOOLEAN FUNCTIONS AND AN ALGORITHM FOR FINDING SHORTEST PATHS IN A GRAPH
Pages: 226 Ph.D. Dissertation/June 1980 $10.10
Keywords: Boolean functions, circuit complexity, Boolean formula size, monotone networks, threshold functions, quadratic functions, algorithms, shortest paths, directed graphs, transitive closure, computational complexity
Abstract: The first part of this thesis considers the complexity of Boolean functions. The main complexity measures used are the number of gates in combinational networks and the size of Boolean formulas. The case of monotone realizations, using only the operations of AND and OR, of monotone functions is emphasized.

For a particular class of monotone functions, the quadratic functions, the worst-case values for the monotone circuit complexity is shown to be proportional to $n^2/\log n$. The number of $\wedge$-gates necessary to compute any quadratic function is also analyzed.

A technique for deriving bounds on monotone circuit size of threshold functions is applied to the "majority" function, (threshold n/2), to establish a lower bound on its monotone circuit complexity of 3n-0(1). For the function "threshold 2", previously known lower bounds on the number of V-gates required are extended in the case of a circuit which has a minimal number of $\wedge$-gates. As a result, it follows that no monotone circuit for "threshold 2" can simultaneously have both a minimal number of $\wedge$-gates and a minimum number of V-gates.

The complexity of combinations of functions on disjoint sets of variables is studied, and a gap between the formula and circuit size of a particular function is given.

Finally, we study the effect of allowing negation in a formula for monotonic functions. Examples are given both of functions in which using negations allows more succinct expressions, and functions in which it does not.

The second part of this thesis describes an algorithm for computing shortest paths in a graph. These results show that an algorithm originally proposed by Spira for this problem can have slow running time. The lacuna in his algorithm is repaired, and it is shown to have $0(n^2(\log n)^2)$ average running time over wide classes of graphs as Spira originally claimed. As a special case, a transitive closure algorithm with $0(n^2 \log n)$ average time is also described.

TR-239
Baker, C.M.
ARTWORK ANALYSIS TOOLS FOR VLSI CIRCUITS
Pages: 75 S.M. & E.E. Thesis/June 1980 $5.60
Keywords: VLSI, artwork analysis, circuit extraction, design rule checking
Abstract: Current methods of designing VLSI chips do not insure that the chips will perform correctly when manufactured. Because the turn around time on chip fabrication varies from a few weeks to a few months, a scheme other than "try it and see if it works" is needed. Checking of chips by hand simulation and visual inspection of checkplots will not catch all of the errors. In addition, the number of transistors per chip is likely to increase from ten thousand to over a million in the next few years. This increase in complexity precludes any manual verification methods; some better method is needed.

A series of programs that use the actual mask descriptors for input are described. These programs perform various levels of checks on the masks, yielding files suitable for simulation. Some of the checks are the usual "design rule" checks of looking for minimum line widths and adequate spacing between the wires. However, there are many more constraints in VLSI circuits than are expressed by the usual design rules. The programs check these constraints using the mask descriptions as input. All of the errors mentioned so far can be classified as syntactic errors; in addition, certain semantic errors are detected. The detection of semantic errors requires various levels of simulation. The input to the simulators is derived from the artwork. {AD A087-040}

TR-240
Montz, L.B.
SAFETY AND OPTIMIZATION TRANSFORMATIONS FOR DATA FLOW PROGRAMS
Pages: 85 S.M. Thesis/July 1980 $5.90
Keywords: data flow programming, data flow translation, optimization, asynchronous systems, Petri nets
Abstract: The *data flow* concept of computation seeks to achieve high performance by allowing concurrent execution of instructions based on the availability of data. This thesis explores the translation of a subset of the high level languages VAL to data flow graphs. The major problem in performing this translation for the target machine, the Dennis-Misunas data flow computer, stems from the restriction that graph execution sequences place at most one value on any given arc at any time. The *data/acknowledge are pair transformation* is introduced as a means of implementing this required operational behavior. Its effect on data flow graph operation is subsequently explored as it relates to correctness and performance.

Through the arc transformation enables graphs to be executed without the possibility of deadlock, the resulting overhead and the potential loss of some concurrency represent significant costs. Two techniques aimed at minimizing these problems are developed for optimizing transformed graphs. The optimization to *eliminate unneeded acknowledge arcs* analyzes VAL constructs to identify arc pairs which permit removal of their acknowledge arc. The optimization to *balance token flow* specifies a method of inserting identity operators into a graph for the purpose of pipelining input sets, and thereby increasing graph throughput. Though developed within the context noted, the translation and optimization issues described should prove applicable to other data flow architectures.

TR-241
Archer, R.F.
REPRESENTATION AND ANALYSIS OF REAL-TIME CONTROL STRUCTURES
Pages: 109 S.M. Thesis/August 1980 $6.60
Keywords: real-time systems, control structures, control flow graph, scheduling, interrupts, latency, codestripping
Abstract: A new notation is introduced for representing real-time scheduling at the task and event level. These schedules are called control are control structures. The primary constructs included which direct the flow of control are sequencing, iteration, and preemption. Additional notation allows the representation of interrupt masking, task termination by external events, task restart as well as resumption from the finding point of preemption structure of a given control structure notation.

The types of representable control structures are classified by the topology of their Control Flow Graphs. It is shown that although branching is allowed in the preemption structure, a tree shaped preemption structure cannot be represented. Both partial and total orderings of tasks and interrupt priorities are supported, however.

A terminology for describing real-time properties of control structures is developed, and it is seen that without certain assumptions about task execution times and event timings, conclusions cannot be drawn regarding real-time performance of a control structure. A series of algorithms is presented which make use of these assumptions, and find values for task execution times in the presence of preemption. The algorithms can analyze control structures containing the principal control features; suggestions are given for further development of algorithms which can analyze any representable control structure. {AD A089-828}

TR-242
Loui, M.C.
SIMULATIONS AMONG MULTIDIMENSIONAL TURING MACHINES
Pages: 46 Ph.D. Dissertation/August 1980 $4.70
Keywords: automata, computational complexity, simulation, Turing machine, multidimensional Turing machine, nondeterminism, time complexity, tape complexity, pebble game, time-space tradeoff, overlap
Abstract: This thesis presents three independent papers: nearly optimal on-line simulations among multidimensional Turing machines, a space bound for one-tape multidimensional Turing machines, and new proofs in the pebble game.

line by a deterministic multihead *d*-dimensional Turing machine in time $O(T(n)^{1+1/d-1/e+\epsilon})$. This simulation almost achieves the known lower bound $\Omega(T(n)^{1+1/d-1/e})$ on the time required.

Every nondeterministic *d*-dimensional Turing machine with one worktape head of time complexity $T(n)$ can be simulated by a deterministic Turing machine of space complexity $(T(n) \log T(n))^{d/(d+1)}$. The proof includes a generalization of crossing sequences.

An overlap argument is used to derive two new proofs in the pebble game. Every directed acylic graph G with n vertices and bounded in degree can be pebbled with $O(n/\log n)$ pebbles. Furthermore, is $S \geq O(n/\log n)$, then G can be pebbled with S pebbles in time $S2^{2^{O(n/S)}}$.

TR-243
Svobodova, L.
MANAGEMENT OF OBJECT HISTORIES IN THE SWALLOW REPOSITORY
Pages: 79 August 1980 $5.70
Keywords: distributed systems, atomicity, storage management, reliability, recovery, SWALLOW
Abstract: **SWALLOW** is an experimental distributed data storage system that provides personal computers with a uniform interface to their local data and the data stored in shared remote servers called repositories. The **SWALLOW** repositories provide reliable, secure, and efficient long-term storage for both very small and very large objects and support updating of a group of objects at one or several repositories in a single atomic action. The repositories support, with some minor modifications, the object model developed by Reed [Reed 78].

The core of the repository is stable *append-only* storage called the Version Storage (VS). VS is the only stable storage in the repository. It contains the histories of all objects in the repository and all the information needed for crash recovery. It is assumed that VS will be implemented with write-once storage devices such as optical disks. The upper 2^n words of VS are kept in the Online Version Storage (OVS). Techniques similar to real-time garbage collection are used to keep current versions of frequently used objects in OVS. Two different policies for retaining current versions of objects in OVS are investigated; the actual implementation further depends on the type of storage devices used for OVS.

A critical concern addressed throughout the design of the repository is recovery from crashes and storage device failures. The crash recovery of the repositories is based entirely on the information contained in VS; VS is scanned sequentially, starting from its current end, until all objects histories have been reconstructed. The recovery can be distributed over time, such that the recovery process is invoked for one object at a time, as individual objects areaccessed. The same mechanism is used to recover *commit records*, which are data structures that record the state of atomic actions and group together the objects to be updated in a single atomic action. The implementation of commit records in the repository guarantees that all updates made by a specific atomic action are either all completed or all undone, regardless of failures. Further, interrupted atomic actions can be continued from the point of interruption, without any additional (backward) recovery. {AD A089-836}

TR-244
Ruth, G.R.
DATA DRIVEN LOOPS
Pages: 97 August 1980 $6.25
Keywords: automatic programming, software design, very high level languages
Abstract: The notion of the data driven loop arises in connection with our work in the Very High Level Language HIBOL and the automatic programming system (ProtoSystem I) that supports it. Although the concept is of general interest outside of VHLL's and automatic programming, we find it profitable to use HIBOL as a vehicle for our discussion and a means of narrowing the scope of our discussion. Therefore, we first present a brief description of the domain which HIBOL treats.

TR-245
Church, K.W.
ON MEMORY LIMITATIONS IN NATURAL LANGUAGE PROCESSING
Pages: 162 S.M. Thesis/September 1980 $8.20
Keywords: parsing, natural language, psychological reality, finite state, determinism, memory limitations, performance
Abstract: This paper proposes a welcome hypothesis: a computationally simple device is sufficient for processing natural language. Traditionally it has been argued that processing natural language syntax requires very powerful machinery. Many engineers have come to this rather grim conclusion; almost all working parsers are actually Turing Machines (TM).

For example, Woods specifically designed his Augmented Transition Networks (ATN's) to be Turing Equivalent. If the problem is really as hard as it appears, then the only solution is to grin and bear it. Our own position is that parsing acceptable sentences is simpler because there are constraints on human performance that drastically reduce the computational requirements (time and space bouds). Although ideal linguistic competence is very complex, this observation may not apply directly to a real processing problem such as parsing. By including performance factors, it is possible to simplify the computation. We will propose two performance limitations, bounded memory and deterministic control, which have been incorporated in a new parser YAP.

TR-246
Tiuryn, J.
A SURVEY OF THE LOGIC OF EFFECTIVE DEFINITIONS
Pages: 47 October 1980 $4.75
Keywords: effective definitions, logics of programs, partial correctness, completeness, infinitary logic
Abstract: LED, the Logic of Effective Definitions, is an extension of first order predicate calculus used for making assertions about programs. Programs are modeled as effective definitional schemes (following Friedman). Logical properties of LED and its relations to classical logics and other programming logics are surveyed.

TR-247
Weihl, W.E.
INTERPROCEDURAL DATA FLOW ANALYSIS IN THE PRESENCE OF POINTERS, PROCEDURE VARIABLES, AND LABEL VARIABLES
Pages: 43 S.B.& S.M. Thesis/October 1980 $4.60
Keywords: optimizing compilers, data flow analysis, procedure variables, P-space hard
Abstract: The compilation of highly modular programs requires extensive interprocedural analysis in order to produce reasonable object code. Such analysis is greatly complicated when the source language contains such constructs as procedure variables and label variables. The possibility of aliasing among variables in the programs to be analyzed adds further complications.

Procedure variables make it impossible to determine, from a simple scan of the program, which procedures may be called by each call statement. Label variables similarly make it impossible to determine which labels may be gone to by each goto statement. Thus a call graph and a control flow graph cannot be constructed after a simple scan of the program. This information is needed to perform intraprocedural data flow analysis.

We suggest an approach to analyzing programs with the features mentioned above, and describe an algorithm which can be used to implement it. The approach involves determining possible values for procedure and label variables and the possible alias relationships among variables in the program. The problem of determining possible values for procedure variables is also shown to be P-space hard. This indicates that the problem is likely to be intractable, and motivates the search for approximate solutions. The algorithm which we propose therefore produces information which is safe but not always precise. It has a running time which is approximately bounded by the product of the number of alias relationships in the program and the number of variables and constants of pointer, procedure or label type. In certain cases it is as precise as possible, and in some of these it is also asymptotically as efficient as possible.

TR-248
LaPaugh, A.S.
ALGORITHMS FOR INTEGRATED CIRCUIT LAYOUT: AN ANALYTIC APPROACH
Pages: 175 Ph.D. Dissertation/November 1980 $8.60
Keywords: VLSI, circuit layout, component placement, channel routing, NP-complete, algorithm analysis, heuristic algorithms
Abstract: In this thesis, the problem of designing the layout of integrated circuits is examined. The layout of an integrated circuit specifies the position of the chip of functional components and wires interconnecting the components. We use a general model under which components are represented by rectangles, and wires are represented by lines. This model can be applied to circuit components defined at any level of complexity, from a transistor to a programmable logic array (PLA). We focus on the standard decomposition of the layout problem into a placement problem and a routing problem.

We examine problems encountered in layout design from the point of view of complexity theory. The general layout problem under our model is shown to be NP-complete. In addition, two problems encountered in a restricted version of the routing problem — channel routing — are shown to be NP-complete. The analyusis of heuristic algorithms for NP-complete problems is discussed, and the analysis of one common algorithm is presented.

The major result presented in this dissertation is a polynomial time algorithm for a restricted case of the routing problem. Given one rectangular component with terminals on its boundary, and pairs of terminals to be connected, the algorithm will find a two-layer channel routing which minimizes the area of a rectangle circumscribing the component and the wire paths. Each terminal can appear in only one pair of terminals to be connected, and the rectangle used to determine the area must have its boundaries parallel to those of the component. If any of the conditions of the problem are removed, the algorithm is no longer guaranteed to find the optimal solution.

TR-249
Turkle, S.
COMPUTERS AND PEOPLE: PERSONAL COMPUTATION
Pages: 38 December 1980 $4.45
Keywords: personal computers, computer culture, computers and society, societal implications of computers
Abstract: Observations made from an ethnographic investigation of the cultures and subcultures around computation,looking at the relationships that people form with computers and with each other in the social worlds that grow up around the machines.

TR-250
Leung, C.K.C.
FAULT TOLERANCE IN PACKET COMMUNICATION COMPUTER ARCHITECTURES
Pages: 169 Ph.D. Dissertation/December 1980 $8.40
Keywords: data flow architecture, self-timed systems, fault tolerance, dynamic redundancy, fault-tolerant networks, fault-tolerant synchronization, non-determinacy
Abstract: It is attractive to implement a large scale parallel processing system as a self-timed hardware system with decentralized control and to improve maintainability and availability in such a system through fault tolerance. In this thesis we show how to tolerate hardware failures in a self-timed hardware system with a packet communication architecture, designed to execute parallel programs organized by data flow concepts.

We first formulate a design methodology for incorporating redundant hardware into self-timed systems for fault tolerance. Redundancy management problems in self-timed systems are illustrated with a byte-sliced hardware module structure. Robust algorithms are given for synchronizing byte slices in a redundant module so that their outputs can be decoded to detect and/or mask hardware failures. Hardware implementation of these redundancy management algorithms is studied under a stuck-at fault model, a random pulse train fault model and a random wave train fault model.

In studying the design of fault-tolerant data flow processors we have also developed a dynamic redundancy scheme for masking hardware failures in a multiprocessor architecture designed to execute parallel programs organized by data flow principles. Novel features of this architecture include use of packet networks to support communication among processing elements and dynamic allocation of a homogeneous set of functional units to service requests. Program organization and hardware module designs to support the dynamic redundancy scheme are described.

TR-251
Swartout, W.R.
PRODUCING EXPLANATIONS AND JUSTIFICATIONS OF EXPERT CONSULTING PROGRAMS
Pages: 117 Ph.D. Dissertation/January 1981 $6.85
Keywords: explanation, automatic programming, expert systems
Abstract: Traditional methods for explaining programs provide explanations by converting to English the code of the program or traces of the execution of that code. While such methods can provide adequate explanations of what the program does or did, they typically cannot provide justifications of the code without resorting to canned-text explanations. That is, such systems cannot tell why what the system is doing is a reasonable thing to be doing. The problem is that the knowledge required to provide these justifications is needed only when the program is being written and does not appear in the code itself. In the XPLAIN system, an automatic programming approach is used to capture some of the knowledge necessary to provide these justifications.

The XPLAIN system uses an automatic programmer to generate the consulting program by refinement from abstract goals. The automatic programmer uses a domain model, consisting of facts about the application domain, and a set of domain principles which drive the refinement process forward. By keeping around a trace of the execution of the automatic programmer it is possible to provide justifications of the code using techniques similar to the traditional methods outlined above. This paper discusses the system described above and outlines additional advantages this approach has for explanation.

TR-252
Arens, G.C.
RECOVERY OF THE SWALLOW REPOSITORY
Pages: 120 S.M. Thesis/January 1981 $6.95
Keywords: distributed data storage system, hash table, recovery, optical disk, computer systems
Abstract. This thesis presents the design of a set of recovery mechanisms for the Swallow repository. Swallow is a distributed data storage system that supports highly reliable long term storage of arbitrary sized data objects with special mechanisms for implementing multi-site atomic actions The Swallow repository is a data storage server that keeps permanent data in write-once stable storage such as optical disk.

The recovery mechanisms provide on-line recovery for the repository's internal data, as the repository proceeds with its normal operations. In this way users that wish to access any data that was not affected by the crash can do so while the damaged data is being recovered. Included in the repository's recovery mechanisms are *recovery epochs* and *checkpoint epochs*, which facilitate the detection of damage to the data and minimize the amount of recovery that is necessary. Also included are specialized hash table algorithms that are immune to repository failures. In addition to describing these mechanisms, this thesis discusses how they support the global recovery mechanisms of Swallow and analyzes how they will affect the repository's general performance. {AD A096-374}

TR-253
Ilson, R.
AN INTEGRATED APPROACH TO FORMATTED DOCUMENT PRODUCTION
Pages: 109 S.M. Thesis/February 1981 $6.60
Keywords: document processing, office automation, text editing
Abstract: Recent advances in printing technology have reduced the cost of typset quality printers. Unfortunately, the production of attractively formatted documents requires typographic skill and special training on computer-based text processing systems. In response to this situation, we have developed the Etude text processing system. The principal characteristics of Etude are that it embodies substantial typographic expertise, and is based on concepts familiar to untrained users. Furthermore, Etude provides a real-time display facility that allows the results of editing and formatting operations to be seen immediately. Thus, Etude supports the entire process of producing decorously formatted documents.

TR-254
Ruth, G.R., Alter, S., Martin, W.A.
A VERY HIGH LEVEL LANGUAGE FOR BUSINESS DATA PROCESSING
Pages: 407 March 1981 $15.55
Keywords: business applications, programming languages
Abstract: The focus of this report is an on-going research effort whose basic purpose is to produce fundamental improvements in the software technology relevant to business applications systems. The basic philosophy of this research is that the system development process can be significantly expedited by the further automation of its part. We give a history of applications systems technology and show how it can be expected to evolve in the future.

TR-255
Kent, S.
PROTECTING EXTERNALLY SUPPLIED SOFTWARE IN SMALL COMPUTERS
Pages: 254 Ph.D. Dissertation/March 1981 $10.95
Keywords: computer security, protection, proprietary software, cryptography, personal computers, distributed systems, data encryption standard, public-key cryptosystems
Abstract: The increasing decentralization of computing resources and the proliferation of personal and small business computers create new problems in computer security. One such problem is the protection of *externally supplied software*, i.e., software supplied by other than the users/owners of these small computers. In the case of personal and small business computers, proprietary software serves as the primary example. In distributed systems comprised of autonomously managed nodes, members of the user community may act as vendors of external software in a less formal context. In these contexts dual security requirements arise: vendors require encapsulation of their software to prevent release and to detect modification of information, whereas users require confinement of external software in order to control its access to computer resources. The protection mechanisms developed to support mutually suspicious subsystems in centralized systems are not directly applicable here because of differences in the computing environment, e.g., the need to protect external subsystems from physical attacks mounted by owners of these small computers.

This thesis employs two tools to achieve the security requirements of vendors of external software: tamper-resistant modules (TRMs) and cryptographic techniques. The former provide physical security, i.e., while the TRM is intact it prevents the release or modification of information contained within and breaking into a TRM results in destruction (erasure) of the sensitive information inside. Packaging all of the sensitive components of a computer system (processor and storage) in a single TRM is often impractical, but selected portions of a system can be protected effectively in this fashion. Cryptographic techniques are employed in two ways in this application: to secure communication among TRMs and to protect information held in physically unprotected storage outside a TRM.

These tools address the problem of encapsulating external software but do not provide the confinement required by users. External software can be confined in two ways: through the use of a secure operating system in conjunction with a TRM supplied by a third-party or by providing separate processors for vendors and users and employing some simple hardware to implement access control for the user. Designing small computer systems incorporating these security features requires careful analysis of a number of options in making tradeoffs among performance, cost, flexibility and security.

TR-256
Faust, G.G.
SEMIAUTOMATIC TRANSLATION OF COBOL INTO HIBOL
Pages: 177 S.M. Thesis/April 1981 $8.65
Keywords: program translation, program understanding, COBOL, HIBOL, automatic programming
Abstract: A severe software crisis is currently being experienced by the data processing community due to intolerable maintenance costs. A system is introduced to reduce those costs by the translation of existing COBOL software into HIBOL; a very high level language that is significantly easier to maintain. HIBOL, uses a single type of data object, called a *flow*, which is an indexed stream of data values. Computation is expressed as operations acting on flows.

The translation process relies on a method for program abstraction developed by Richard Waters which expresses programs as a hierarchical structure, called an analyzed plan, in which control and data flow is made explicit. In this formalism loops are expressed as a composition of stream operators acting on stream data flow.

This paper discusses in detail how an analyzed plan for a COBOL program can be translated into a HIBOL program. It is currently possible to translate into HIBOL analyzed plans for a relatively small (but well defined) subset of COBOL programs. Suggestions are made as to how that subset could be expanded through further research.

TR-257
Cesari, C.A.
APPLICATION OF DATA FLOW ARCHITECTURE TO COMPUTER MUSIC SYNTHESIS
Pages: 129 S.B./S.M. Thesis/February 1981 $7.20
Keywords: sound synthesis, voltage waveform, hardware synthesizers, real-time performance, orchestra file, score file, data flow, VAL, streams, pipelining
Abstract: A computer music synthesis system is the most flexible of synthesis systems. It offers a composer extensive control over the sound of his piece. A user of such a system describes his composition in some synthesis language. The computer uses this description to calculate samples of a voltage waveform that can be fed to D/A converters at a specified sampling rate. The D/As' outputs are in turn fed to loudspeakers that produce the sound of the user's composition. Real time performance is unattainable on existing computer synthesis systems due to the sequential nature of conventional computers. Unless the parallelism that is present in

the sample calculation process is exploited, real time performance will remain unobtainable. This thesis presents a proposed computer synthesis system that includes a data flow machine, a computer whose architecture is highly parallel. The Music-11 synthesis system at MIT was used as a model in its design. An analysis of the algorithms used in the sample conversion process and how it would run on the data flow machine is presented. An example of how a composition would be described in a synthesis language and how it would run on the proposed system is given.

TR-258
Singh, N.P.
A DESIGN METHODOLOGY FOR SELF-TIMED SYSTEMS
Pages: 98 S.M. Thesis/February 1981 $6.25
Keywords: self-timed systems, speed independent, asynchronous design methodology, VLSI
Abstract: This thesis presents a design methodology for self-timed systems which will be extremely attractive for implementing systems in VLSI. Self-timed systems are characterized by the absence of a timing reference to which all operations are synchronized. Currently most systems are implemented using a synchronous design methodology where all operations are synchronized to a global clock. However, this approach will not be attractive in the future for implementing systems in VLSI due to the high communication costs in VLSI and the prohibitive task of managing timing constraints global to a VLSI integrated circuit.

The methodology proposed in this thesis defines a set of modules which form the building blocks for implementing an arbitrary self-timed system. The various module types are based on familiar programming constructs such as iterations, conditionals and constructs that aid in the activation and synchronization of parallel processes, such as, forking and joining. The modules of a self-timed system communicate with each other via an asynchronous communication protocol, and the correct behavior of the system is independent of the delays in the communication medium. This methodology simplifies the design effort by restricting the timing constraints to be local to the modules of the system.

TR-259
Bryant, R.E.
A SWITCH-LEVEL SIMULATION MODEL FOR INTEGRATED LOGIC CIRCUITS
Pages: 219 Ph.D. Dissertation/March 1981 $9.90
Keywords: switch-level simulation, logic simulation, computer-aided design, VLSI
Abstract: The switch-level model describes the logical behavior of digital integrated circuits implemented in metal oxide semiconductor (MOS) technology. A network in this model consists of a set of nodes connected by transistor "switches." Many aspects of MOS circuits can be described which cannot be expressed in the Boolean logic gate model, such as bidirectional pass transistors, dynamic storage, and charge sharing. Furthermore, the logic network can be extracted directly from the mask specification of a circuit by a relatively straightforward computer program. Unlike analog circuit models, however, the nodes in a switch-level network assume discrete logic states 0, 1, and X (for unknown), and the transistors assume discrete states "open," "closed," and "unknown." This model can form the basis of a logic simulator for MOS circuits with performance comparable to logic gate simulators. This dissertation presents a rigorous development of the switch-level model and several simulation algorithms.

TR-260
Moss, E.B.
NESTED TRANSACTIONS: AN APPROACH TO RELIABLE DISTRIBUTED COMPUTING
Pages: 178 Ph.D. Dissertation/April 1981 $8.65
Keywords: distributed computing, reliability, fault tolerance, transaction
Abstract: This report addresses the issue of providing software for reliable distributed systems. In particular, we examine how to program a system so that the software continues to work in the face of a variety of failures of parts of the system. The design presented uses the concept of transactions: collections of primitive actions that are indivisible. The indivisibility of transactions insures that consistent results are obtained even when requests are processed concurrently or failures occur during a request. Our design permits transactions to be nested. Nested transactions provide nested universes of synchronization and recovery from failures. The advantages of nested transactions over single-level transactions are that they provide concurrency control within transactions by serializing subtransactions appropriately, and that they permit parts of a transaction to fail without necessarily aborting the entire transaction. The method for implementing nested transactions described in this report is novel in that it ues locking for concurrency control. We present th e necessary algorithms for locking, recovery, distributed commitment, and distributed deadlock detection for a nested transaction system. While the design has not been implemented, it has been simulated. {AD A100-754}

TR-261
Martin, W.A., Church, K.W., Patil, R.S.
PRELIMINARY ANALYSIS OF A BREADTH-FIRST PARSING ALGORITHM: THEORETICAL AND EXPERIMENTAL RESULTS
Pages: 86 June 1981 $5.90
Keywords: parsing, chart parsing, natural language, Early's algorithm
Abstract: We will trace a brief history of context-free parsing algorithms and then describe some representation issues. The purpose of this paper is to share our philosophy and experience in adapting a well-known context free parsing algorithm (Earley's algorithm and variations thereof) to the parsing of a difficult and wide ranging corpus of sentences. The sentences were gathered by Malhotra in an experiment which fooled businessmen users into thinking they were interacting with Malhotra in another room. The Malhotra corpus is considerably more difficult than a second collection published by the LADDER Group. Both collections are given in the appendices. Section 4 compares empirical results obtained from these collections against theoretical predictions.

TR-262
Todd, K.W.
HIGH LEVEL VAL CONSTRUCTS IN A STATIC DATA FLOW MACHINE
Pages: 74 S.M. Thesis/June 1981 $5.55
Keywords: VAL, static data flow machine, instruction cells, pipelining, sharing, streams
Abstract: The Dennis-Misunas Form 1 Data Flow Machine can best be described as a static and scalar machine. Despite these two limiting characteristics, it is still possible to translate the whole of the functional programming language VAL into the base language of this machine. Methods for translating the various high level constructs of VAL are presented which exploit the parallelism inherent in programs written in VAL mainly by pipelining through a single expression (vertical parallelism) rather than employing many copies of that same expression (horizontal parallelism), although the latter is not ruled out. These methods are tested by translating two different versions of a vector dot product algorithm, and the results obtained from running these translations on an interpreter are analyzed.

TR-263
Street, R.S.
PROPOSITIONAL DYNAMIC LOGIC OF LOOPING AND CONVERSE
Pages: 37 Ph.D. Dissertation/May 1981 $4.45
Keywords: logics of programs, dynamic logic, propositional dynamic logic, regular programs, infinite loops, decision method, finite automata, tree automata
Abstract: Propositional dynamic logic is a formal system for reasoning about the before-after behavior of regular program schemes. This thesis investigates extensions of this system; in particular, propositional dynamic logic is extended to include, first, an infinite looping construct, and second, both an infinite looping construct and a converse or backtracking construct. The focus is on showing that the extensions express properties of programs inexpressible in the original logic and on establishing upper bounds on the computational complexity of the satisfiability problems for the new logics.

In order to establish the decidability of propositional dynamic logic of looping, a special class of finite automata on infinite trees is defined. The emptiness problem for this class is shown to be exponentially easier than the general problem. The satisfiability problem for the extended logic is then shown to be reducible to this special emptiness problem; the result is a triply exponential time decision procedure. The notion of a finitely generable infinite tree is used to prove that the logic satisfies the finite model property, i.e., every satisfiable formula has a finite model.

Propositional dynamic logic of looping and converse does not satisfy the finite model property; there is a formula satisfied in an infinite model, but not in any finite model. The failure of a logic to satisfy the finite model property is often taken as good evidence for its undecidability. Nevertheless, propositional dynamic logic of looping and converse is decidable. In order to establish this result, deterministic two-way automata on infinite trees are defined, and it is shown how they can be simulated by nondeterministic one-way automata on infinite trees. The satisfiability problem for propositional dynamic logic of looping and converse is shown to be reducible to the emptiness problem for these two-way automata; the result is an octuply exponential time decision procedure. The notion of a finitely generable infinite tree is used to prove that every satisfiable formula has a (possible infinite) model with a finite representation. This clarifies why the logic is decidable.

TR-264
Schiffenbauer, R.D.
INTERACTIVE DEBUGGING IN A DISTRIBUTED COMPUTATIONAL ENVIRONMENT
Pages: 187 S.M. Thesis/August 1981 $8.95
Keywords: distributed systems, debugging, monitors, reproducibility, lurking bugs
Abstract: This thesis describes an implementation of a facility for interactively debugging distributed programs. These distributed programs consist of groups of cooperating processes concurrently executing on an arbitrarily extensive network of processors. The facility allows the user to monitor and control, at his leisure, the interprocess communications that occur through message passing while execution of the distributed program proceeds. It presents the user with the ability to simulate transmission errors and delays, to alter and create packets, and to precisely control the pattern of such communications. The facility serves as a tool for the detection of *lurking bugs,* those errors, peculiar to parallel processing, which may or may not appear during the course of any particular execution.

The facility possesses a high degree of *transparency* towards the program being debugged. That is, it has a minimal effect on the events that define the execution of that program. Transparency is a desirable property for any debugger to possess. To achieve such transparency, the processes of the distributed program are made to execute in a logical time environment, reading logical, rather than physical, clocks.

We show that the facility obeys a *clock condition,* with which any logical time system must comply in order to be correct. We also show that the facility actually *simulates* the program it is being used to debug. Finally, we show that the facility simulates a particular computation of the program that is likely to occur. The notion of *probable simulation* is defined, and our debugging facility is shown to achieve it.

TR-265
Thomas, R.E.
A DATA FLOW ARCHITECTURE WITH IMPROVED ASYMPTOTIC PERFORMANCE
Pages: 82 Ph.D. Dissertation/April 1981 $5.80
Keywords: data flow architecture, architecture
Abstract: Large scale integration presents a unique opportunity to design a computer compromising large numbers of small, inexpensive processors. This paper presents a design for such a machine based on the asynchronous and functional semantics of data flow. Processors within the machine are interconnected by a packet-switched binary n-cube although a limited number of other networks may be substituted with predictable asymptotic effects on performance. Improved performance of the proposed machine over a previously reported data flow architecture is predicted in terms of the computational time complexity of several example programs: matrix multiply, quicksort, and iterative solutions to partial differential equations. Although the example programs are numerical in nature, the machine is intended for general-purpose computation since programs are written in the high level data flow language Id without knowledge of the number of processors or interconnections. New storage management and data communication methods are also presented which are necessary to obtain the improved performance. Experimental results from a simulated machine incorporating some of these methods are given to corroborate analytic results.

TR-266
Good, M.
AN EASE OF USE EVALUATION OF AN INTEGRATED EDITOR AND FORMATTER
Pages: 153 S.M. Thesis/August 1981 $7.90
Keywords: human factors, man computer interface, office automation, text editing
Abstract: Etude is an integrated text editor and formatter that was designed to be easy to learn and easy to use. To measure Etude's success in meeting these goals, twenty-ome computer-naive temporary office workers were taught to use Etude in a controlled experiment. Ninety percent of the subjects were able to create and edit letters after a training period of less than two hours and twenty minutes, though they were not able to perform these tasks as quickly as they could when using a typewriter. Etude did not appear to have any systematic effect on subject anxiety. The subjects had favorable attitudes towards using Etude. These attitudes were at least as favorable as their attitudes towards using a typewriter.

TR-267
Patil, R.S.
CAUSAL REPRESENTATION OF PATIENT ILLNESS FOR ELECTROLYTE AND ACID-BASE DIAGNOSIS
Pages: 137 Ph.D. Dissertation/October 1981 $7.45
Keywords: causal representation, medical diagnosis, clinical decision making
Abstract: Much of the medical knowledge in the first generation *AI in Medicine* programs is phenomenological; that is, it describes the associations among phenomena without knowledge of the underlying causal mechanisms. Although these AIM programs provide a good first approximation to the way clinicians reason, they fail to reproduce clinicians' reasoning based on a deeper understanding of the phenomena. More specifically, they do not deal with the knowledge of disease at different levels of detail, nor do they utilize causal relations to organize and explain the clinical facts and disease hypothesis. They also cannot deal with illnesses resulting from multiple diseases, especially when one disease alters the presentation of the others. Finally, they are unable to capture the notions of adequacy and parsimony that play such a large role in diagnosis. To explore these issues and rectify these deficiencies, we have undertaken the task of providing expert consultation for electrolyte and acid-base disturbances.

This thesis reports the implementation of *ABEL,* the diagnostic component of the consultation program. In it, we explore the problems of modeling the causal understanding of a patients illness. We develop techniques for dealing with illness resulting from multiple interacting diseases. We describe a multi-level representation of causal knowledge, and explore issues of the aggregation of available case specific knowledge into concise summaries of the patients illness. We discuss structural criteria for evaluating parsimony, coherence and adequacy of diagnostic explanations. We also explore some of the issues involved in information gathering and propose expectation-driven diagnostic planning as a means of improving it. Finally, we discuss the issues of explanation and justification of the program's understanding and argue that these facilities are crucial for acceptability of a consultation program.

TR-268
Guttag, J.V., Kapur, D., Musser, D.R.
DERIVED PAIRS, OVERLAP CLOSURES, AND REWRITE DOMINOES: NEW TOOLS FOR ANALYZING TERM REWRITING SYSTEMS
Pages: 19 December 1981 $3.90
Keywords: confluence, critical pair, derived pair, Knuth-Bendix, overlapclosure, restricted termination, rewrite dominoes, term rewriting systems, uniform termination
Abstract: Starting from the seminal work of Knuth and Bendix, we develop several notions useful in the study of term rewriting systems. In particular we introduce the notions of "derived pairs" and "overlap closure" and show that they are useful in analyzing sets of rewrite rules for various properties related to termination. We also introduce a new representation, based on rewrite dominoes, for rewrite rules and sequences of rewrites.

TR-269
Kanellakis, P.C.
THE COMPLEXITY OF CONCURRENCY CONTROL FOR DISTRIBUTED DATA BASES
Pages: 113 Ph.D. Dissertation/December 1981 $6.70
Keywords: concurrency control, distributed data base, communication, complexity, PSPACE-complete, locking
Abstract: This study is an analysis of the distributed version of data base concurrency control. It provides concrete mathematical evidence that the distributed problem is an inherently more complex task than the centralized one.

The notions of transaction, concurrency, history, serializability, scheduler, etc., for centralized data bases are now well-understood both from a theoretical and a practical point of view. A formal model for the case of distributed data bases is presented. The transactions are partially ordered sets of actions, as opposed to the totally ordered straight-line programs of the centralized case. The scheduler is also a distributed program. Three notions of performance for a scheduler are studied and interrelated: (i) parallelism, (ii) the computational complexity of the decision problems that it has to solve, (iii) the cost of communication between the various parts of the scheduler. In fact the number of messages necessary and sufficient to support a given level of parallelism is equal to the length of a combinatorial game. This game, which captures the difference between the centralized and the distributed problem, is PSPACE-Complete. This implies that unless NP = PSPACE, a scheduler cannot simultaneously minimize the communication cost and be computationally efficient.

The model presented can also serve as a framework for the study of distributed concurrency control by locking. For two transactions an efficient characterization of safe distributed locking policies is derived. The new graph-theoretic approach generalizes the geometric method used in the centralized case.

TR-270
Singh, V.
THE DESIGN OF A ROUTING SERVICE FOR CAMPUS-WIDE INTERNET TRANSPORT
Pages: 109 S.M. Thesis/January 1982 $6.60
Keywords: computer networks, routing, servers
Abstract: A campus-wide network requires many subnetworks connected by gateways and it has a relatively loose administration. Modularization of network implementation is important in this environment to make efficient use of ever-improving technologies and protocols. The need for modularization makes it desirable to separate a routing and target identification scheme from gateway implementation - a facility that source routing provides. Moreover, removing routing and target identification responsibilities from the gateways leads to their simplicity and, therefore, a better chance that gateways will not be bottlenecks in the high-bandwidth network. This thesis focuses on the design of a Routing Service to support source routing in the campus environment.

The Routing Service is designed to find paths from a requesters attachment point to a node specified by the requester. The Routing Service accepts hints from the requester about the destination node's location in the network to limit the search involved. The Routing Service also provides user-control of paths and diagnosis for faulty paths. The design of the Routing Service places strong emphasis on scaleability with respect to the size of the network. Reliability and simplicity are two other key features of the Routing Service.

TR-271
Rutherford, C.J., Davies, B., Barnett, A.I., Desforges, J.F.
A COMPUTER SYSTEM FOR DECISION ANALYSIS IN HODGKINS DISEASE
Pages: 115 February 1982 $6.80
Keywords: decision-support system, decision analysis, Hodgkins disease, threshold analysis, Bayes Rule, medical diagnosis, clinical decision making
Abstract: This report describes the development of an interactive computer system to assist physicians in evaluating and managing patients with Hodgkins disease. The system employs the techniques of decision analysis to select a sequence of testing and treatment which optimizes the patients chance of long-term survival. The system additionally presents the physician user with information about patient discomfort and dollar cost to facilitate choice between alternatives with approximately equal survival potential. The statistical data used in the decision analysis is derived from a data base representing approximately 1200 Hodgkins disease patients. The validity of the approach and the underlying assumptions was tested statistically. The methodology of threshold analysis and applications to general management dilemmas in Hodgkins disease are presented. Details of the system design and implementation in the Lisp programming language are shown. Much of the statistical data used by the program is included as an appendix.

TR-272
Smith, B.C.
REFLECTION AND SEMANTICS IN A PROCEDURAL LANGUAGE
Pages: 495 Ph.D. Dissertation/January 1982 $18.20
Keywords: abstract data types, applicative languages, artificial intelligence, category alignment, closures, continuations, control structures, debugging, dynamic scoping, environments, evaluation, first order logic, functional programming, higher order, intention, interpretation, knowledge representation, lambda calculus, LISP, metacircular interpreter, normal-form, procedural reflection, PROLOG, reduction, reflection, SCHEME, self-reference, semantics, side effects simplification, static scoping, structured programming, tail-recursion
Abstract: We show how a computational system can be constructed to "reason", effectively and consequentially, about its own inferential processes. The analysis proceeds in two parts. First, we consider the general question of computational semantics, rejecting traditional approaches, and arguing that the *declarative* and *procedural* aspects of computational symbols (what they stand for, and what behavior they engender) should be analyzed *independently*, in order that they may be coherently related. Second, we investigate *self-referential* behavior in computational processes, and show how to embed an effective procedural model of a computational calculus within that calculus (a model not unlike a meta-circular interpreter, but connected to the fundamental operations of the machine in such a way as to provide, at any point in a computation, fully articulated descriptions of the state of that computation, for inspection and possible modification). Our claims are three: a) rationalized semantics better captures our collective tacit understanding of computation than do previous analyses, b) it is straightforward to add reflective capabilities to a semantically rationalized formalism, and c) the architecture we adopt will support arbitrary computational processes able to shift smoothly between dealing with a given subject domain, and dealing with their own reasoning processes over that domain.

An instance of the general solution is worked out in the context of an applicative language. Specifically, we present three successive dialects of LISP: 1-LISP, a distillation of current practice, for comparison purposes; 2-LISP, a dialect constructed in terms of our rationalized semantics, in which the concept of evaluation is rejected in favor of independent notions of *simplification* and *reference,* and in which the respective categories of notation, structure, semantics and behavior are strictly aligned; and 3-LISP, an extension of 2-LISP endowed with reflective powers.

TR-273
Estrin, D.L.
DATA COMMUNICATIONS VIA CABLE TELEVISION NETWORKS: TECHNICAL AND POLICY CONSIDERATIONS
Pages: 155 S.M. Thesis/May 1982 $8.00
Keywords: cable television, telecommunications, computer networks
Abstract: Cable television networks offer peak communication data rates that are orders of magnitude greater than the telephone local loop. Although one-way television signal distribution continues to be the primary application of cable television systems, the cable television network can be used for two-way data communications.

Data communication places severe engineering demands on the performance of a cable television network. Therefore, to ensure that data communications capabilities are not precluded by poor engineering, local cable authorities and the cable industry must identify and overcome the technical barriers to the application of cable television networks to data communications. We identify the following as the primary technical requirements that remain to be addressed by the cable industry:

-Methods for controlling the accumulation of insertion noise and ingress on upstream channels.

-Reliability and security mechanisms to provide adequate levels of system availability, overall quality of service, and privacy of communications.

-System engineering that supports data communications among many nodes, other than on a point-to-point basis.

If the cable industry applies the resources necessary to satisfy these requirements, cable television networks can gain a lucrative share of the growing residential and institutional, data communications and information services markets, in competition with telephone company and cellular microwave services.

In order to make this important, and unique, municipal resource widely available to a diversity of users and service providers, local cable authorities, in addition to the cable industry, should establish structural mechanisms to eliminate the cable operator's potential conflict of interest between its carriage and content functions, which might otherwise inhibit this diversity. The nature of appropriate regulatory mechanisms has been a source of conflict and confusion because of the unclear status of the cable operator as broadcaster or common carrier. Leased channel access requirements, which enforce limited separations on a channel by channel basis, can allow the operator to serve in both capacities, thereby both encouraging investment in facilities and limiting the opportunities for anti-competitive practices. The rate structures and levels adopted by the operator will determine the extent to which leased channel access successfully achieves this end.

More than in the past, cable authorities will find themselves monitoring and specifying details of the network infrastructures, both technical and operational, to ensure that suitable facilities are constructed and that a diversity of service sources are granted access to the medium. This thesis is intended as a resource for cable authorities, as well as cable operators, as they establish the role of cable television networks in the data communications market. In particular chapter 6 serves as a guide for city and state cable authorities to incorporate the technical detail and regulatory structures necessary for data communication services into existing cable policy.

TR-274
Leighton, F.T.
LAYOUTS FOR THE SHUFFLE-EXCHANGE GRAPH AND LOWER BOUND TECHNIQUES FOR VLSI
Pages: 105 Ph.D. Dissertation/August 1981 $6.50
Keywords: crossing number, graph separator, layout area, maximum wire length, parallel computation, shuffle-exchange graph, Thompson model, VLSI
Abstract: The thesis is divided into two parts. In the first part, we describe and analyze several new VLSI layouts for the shuffle-exchange graph. These include:

1) an asymptotically optimal, $\Theta(N^2/\log^2 N)$-area layout for the N-node shuffle-exchange graph, and

2) several practical layouts for small shuffle-exchange graphs.

The new layouts require substantially less area than previously known layouts and can serve as the basis for designing large scale shuffle-exchange chips.

In the second part of the thesis, we develop general methods for proving lower bounds on the layout area, crossing number, bisection width and maximum edge length of VLSI networks. Among other things, we use these methods to find:

1) an N-node planar graph which has layout area $\Theta(N\log N)$ and maximum edge length $\Theta(N^{1/2}/\log^{1/2} N)$,

2) an N-node graph with an $O(N^{1/2})$-separator which has layout area $\Theta(N\log^2 N)$ and maximum edge length $\Theta(N^{1/2}\log N/\log\log N)$, and

3) an -node graph with an $O(N^{\alpha})$-separator (for $\alpha > 1/2$) which has maximum edge length $\Theta(N^{\alpha})$.

The area results indicate that some graphs with $O(N^{1/2})$-separators (and, in particular, some planar graphs) do *not* have linear-area layouts, thus disproving a popular conjecture. The edge length bounds indicate that the layouts of some networks must have very long wires (possibly as long as the width of the layout).

TR-275
Kunin, J.S.
ANALYSIS AND SPECIFICATION OF OFFICE PROCEDURES
Pages: 232 Ph.D. Dissertation/February 1982 $10.30
Keywords: office automation, office analysis, systems analysis, specification languages, integrated office systems
Abstract: Conventional approaches to "office automation" focus on the lowest common denominator of office work: typing, filing, filling in forms, etc. As a consequence, the process of office systems analysis lacks tools and techniques that address the office in terms of business functions rather than as manipulation of paper artifacts. The Office Specification Language (OSL) and its associated analysis methodology have been developed as a means of implementing a *functional* approach to office procedure analysis and description.

OSL is based on several premises derived from a study of office systems analysis at a functional level:

-There exist high-level constructs common to a wide variety of disparate offices. A structured formal language built upon such standardized abstractions can be useful in helping an analyst approach, understand, and describe the operations of many offices.

-Office procedures deal with (abstract) *objects*, not paper forms. Forms and other documents are not basic to office operations: they are mechanisms for organizing and transmitting information about some more fundamental object. Therefore office analysis should focus not on forms, but rather on the underlying business requirements that must survive any change in system implementation.

-Office procedures are fundamentally simple; their apparent complexity is not inherent, but due to a myriad of special cases, historical accretions, and implementation details. identification of a procedure's core requirements is the framework upon which analysis should be based. such an understanding is a prerequisite to effective reorganization of and design of support systems for office functions.

OSL is postulated to be of utility for office analysis and systems design. field tests of the language and methodology have shown that our basic approach is effective for analysis purposes, and have identified directions for further improvements.

TR-276
Srivas, M.K.
AUTOMATIC SYNTHESIS OF IMPLEMENTATIONS FOR ABSTRACT DATA TYPES FROM ALGEBRAIC SPECIFICATIONS
Pages: 161 Ph.D. Dissertation/June 1982 $8.15
Keywords: abstract data types, algebraic specification, association specification, preliminary implementation, term rewriting systems, reduction, expansion
Abstract: Algebraic specifications have been used extensively to prove properties of abstract data types and to establish the correctness of implementations of data types. This thesis explores an automatic method of synthesizing implementations for data types from their algebraic specifications.

The inputs to the synthesis procedure consist of a specification for the implemented type, a specification for each of the implementing types, and a formal description of the representation scheme to be used by implementation. The output of the procedure consists of an implementation for each of the operations of the implemented type in a simple applicative language.

The inputs and the output of the synthesis procedure are precisely characterized. A formal basis for the method employed by the procedure is developed. The method is based on the principle of reversing the technique of proving the correctness of an implementation of a data type. The restrictions on the inputs, and the conditions under which the procedure synthesizes an implementation successfully are formally characterized.

TR-277
Johnson, M.G.
EFFICIENT MODELING FOR SHORT CHANNEL MOS CIRCUIT SIMULATION
Pages: 91 S.M. Thesis/August 1982 $6.05
Keywords: MOS transistor modeling, numerical optimization, nonlinear parameter estimation
Abstract: Existing circuit models for short-channel MOS transistors represent a compromise between speed and ease of use. Empirical models are very fast to evaluate, but their parameters must be fitted from experimental measurements. Theoretical models require longer computation time, but they may be used to predict the performance of new, unmeasured MOS technologies since their parameters are not curve-fitted from experimental data.

This thesis combines the best features of both types of model, yielding a fast circuit simulator whose input parameters need not be extracted from experimental measurements. A nonlinear optimization algorithm is used to "compile" the parameters of a theoretical model into into parameters for an empirical model, providing the superior user-interface of theoretical models without sacrificing simulator execution speed. Results produced by a prototype model compiler are presented, showing the modeling error to be approximately 5 percent.

TR-278
Rosenstein, L.S.
DISPLAY MANAGEMENT IN AN INTEGRATED OFFICE
Pages: 55 S.M. Thesis/January 1982 $5.00
Keywords: display management, man computer interface, office automation
Abstract: Advances in technology now make it possible to build office workstations that have a large amount of local computing power and high-resolution output devices. Such workstations can be used for various office applications, such as document preparation, personal data bases, and electronic mail. In developing applications for such an office workstation, it is desirable to have a common software foundation upon which to build. This not only reduces development time, but also helps to integrate the various applications. Integration, in turn, helps to make the subsystems easy to use and easy to learn.

One component of such a software foundation is a *display manager*, which is used to organize information on the screen. This report describes the design and implementation of the display manager for a software foundation called Ecole. The Ecole display manager provides output services to application programs at three levels: (1) primitive output operations, (2) mechanisms for organizing information on the display, and (3) common display functions.

TR-279
Anderson, T.L.
THE DESIGN OF A MULTIPROCESSOR DEVELOPMENT SYSTEM
Pages: 122 S.M. Thesis/September 1982 $7.00
Keywords: multiprocessor systems, segmented computer buses, parallel processing, computer architecture
Abstract: A multiprocessor development system has been designed an d a prototype system is being constructed. The system, known as Concert, is intended to support multiprocessor research efforts at M.I.T. The motivation for Concert and the project history are summarized briefly. Some intended applications are also identified.

The system incorporates the RingBus architecture, a novel scheme for interconnecting processors and memory in a tightly-coupled multiprocessor system. The architecture is described both in its general form and in the particular implementation used in the system . The results of some analysis and synthesis of the architecture are summarized.

The design of the Concert multiprocessor development system is described, with particular emphasis on the tradeoffs considered in the design process. The design of two particular hardware modules is discussed in considerable detail. Finally, some suggestions are offered for future use of the system and further investigation into the RingBus architecture.

TR-280
Guang-Rong, G.
AN IMPLEMENTATION SCHEME FOR ARRAY OPERATIONS IN STATIC DATA FLOW COMPUTERS
Pages: 85 S.M. Thesis/May 1982 $5.90
Keywords: parallel processing, flow dependency graph, pipelining, data flow architecture, functional programming, VAL
Abstract: The mapping of array operations in VAL programs on a static data flow machine with array memory is studied. The flow dependency graph is introduced as a model of array operations in VAL programs. The balancing and optimization of the flow dependency graphs is presented. The class of well-behaved VAL programs which can be modeled by flow dependency graphs is specified. Schemes for pipelined mapping of **forall** and **for-iter** array operation constructs in well-behaved VAL programs are formulated.

TR-281
Lynch, N.A.
MULTILEVEL ATOMICITY - A NEW CORRECTNESS CRITERION FOR DATA BASE CONCURRENCY CONTROL
Pages: 25 August 1982 $4.10
Keywords: atomicity, concurrency control, data base, serializability, transaction
Abstract: "Multilevel atomicity", a new correctness criteria for data base concurrency control, is defined. It weakens the usual notion of serializability by permitting controlled interleaving among transactions. It appears to be especially suitable for applications in which the set of transactions has a natural hierarchical structure based on the hierarchical structure of an organization. A characterization for multilevel atomicity, in terms of absence of cycles in a dependency relation among transaction steps, is given. Some remarks are made concerning implementation.

TR-282
Fischer, M.J., Lynch, N.A., Paterson, M.S.
IMPOSSIBILITY OF DISTRIBUTED CONSENSUS WITH ONE FAULTY PROCESS
Pages: 10 September 1982 $3.65
Keywords: asynchronous systems, consensus, distributed computing, distributed data base, fault-tolerance
Abstract: The consensus problem involves an asynchronous system of processes, some of which may be unreliable. The problem is for the reliable processes to agree on a binary value. We show that every protocol for this problem has the possibility of nontermination, even with only one faulty process. By way of contrast, solutions are known for the synchronous case, the "Byzantine Generals" problem.

TR-283
Sherman, H.B.
A COMPARATIVE STUDY OF COMPUTER-AIDED CLINICAL DIAGNOSIS
Pages: 139 S.M. Thesis/January 1981 $7.50
Keywords: medical diagnosis, expert systems, birth defects, clinical decision making
Abstract: In recent years many computer systems have been developed to assist in medical decision making. Two of these systems in particular, INTERNIST and the Present Illness Program (PIP), have been proposed as suitable for performing general medical diagnosis. However, there has been no way of comparing the performance of these two programs since the medical data used by the programs differs extensively.

In order to make such a comparison versions of both systems have been implemented, and the medical data used by each has been abstracted from a single data base in the domain of birth defects. Although both systems use a common paradigm of constructing diagnostic hypotheses and then testing those hypotheses by suggesting further tests, variations in their implementation of this paradigm result in significant differences in performance. A detailed analysis of the strengths and weaknesses of these two approaches to computer-aided medical diagnosis, in the diagnosis of thirty-five clinical cases drawn from the congenital defects domain, is presented. The results of this analysis are used to generate suggestions for the improvement of such programs.

TR-284
Cosmadakis, S.S.
TRANSLATING UPDATES OF RELATIONAL DATA BASE VIEWS
Pages: 44 S.M. Thesis/February 1983 $4.65
Keywords: relational data base, view update, view complement, projectiveview, functional dependency, polynomial-time
Abstract: We study the problem of translating updates of data base views. We disambiguate a view update by requiring that a specified *view compliment* (i.e. a second view which contains all the data base information omitted from the given view) remains constant during the translation. We study some of the computational problems related to the application of this general methodology in the context of relational data bases.

We consider, for the most part, data bases consisting of a single relation, with functional dependencies as the only integrity constraints; we also restrict our attention to view defined by projections. We first give a characterization of complementary views (valid also in the presence of join dependencies), which leads to efficient algorithms for checking if two given views are complementary and for determining a non-redundant complement of a given view. We also show that the problem of finding a minimum complement of a given view is NP-complete.

We then study in detail the problem of translating the insertion of a tuple into a view. We show how to do the translation in case the insertion is translatable, and we also develop a polynomial time algorithm for testing translatability; we also give two stronger, more efficient translatability tests. We show lower bounds for the complexity of the translatability problem, by proving that it becomes π_2^p-hard if the view is given in an exponentially succinct way; an analogous result is shown for one of the stronger tests. We also examine the problem of determining a complement which renders a given insertation translatable; we find that it can be solved in time polynomial in the view, but becomes NP-hard if the view is given in an exponentially succinct way; again, analogous results are valid for the stronger tests.

The above results are extended, in a straightforward way, to the cases of deletion and replacement of a tuple. Finally, we define and study a new kind of functional dependencies which is important in the context of complements, the explicit functional dependencies, (EFD's), which intuitively state that some part of the data base information can be computed from the rest. We examine the interaction of EFD's with functional dependencies and join dependencies, and we also extend our characterization of complementary views to allow for the presence of EFD's.

TR-285
Lynch, N.A.
CONCURRENCY CONTROL FOR RESILIENT NESTED TRANSACTIONS
Pages: 31 February 1983 $4.25
Keywords: atomicity, concurrency control, recovery, serializability, transaction, two-phase locking
Abstract: A formal framework is developed for proving correctness of algorithms which implement nwsted transactions. In particular, a simple "action tree" data structure is defined, which describes the ancestor relationships among executing transactions and also describes the views which different transactions have of the data. A generalization of "serializability" to the domain of nested transactions with failures, is defined. A characterization is given for this generalization of serializability, in terms of absence of cycles in an appropriate dependency relation on transactions. A slightly simplified version of Moss' locking algorithm is presented in detail, and a careful correctness proof is given.

The style of correctness proof appears to be quite interesting in its own right. The description of the algorithm, from its initial specification to its detailed implementation, is presented as a series of "event-state algebra" levels, each of which "simulates' the previous one in a straightforward way.

TR-286
Goree, J.A.
INTERNAL CONSISTENCY OF A DISTRIBUTED TRANSACTION SYSTEM WITH ORPHAN DETECTION
Pages: 175 S.M. Thesis/January 1983 $8.60
Keywords: concurrency control, orphans, transaction, serializability, internal consistency
Abstract: This thesis defines a property called "view-serializability", which formalizes internal consistency for a system of nested atomic transactions. Internal consistency is a stronger condition than the usual notion of data base consistency, because it takes into account the views of transactions which will never commit. In a distributed system, local aborts of remote subactions and crashes of nodes can generate *orphans*: active actions which are descendants of actions that have aborted or are guaranteed to abort. Because it is not always feasible or efficient to eliminate orphans immediately, special care is needed to insure that they see consistent system states when they are allowed to continue running. We investigate a particular dynamic detection strategy designed to detect orphans before they violate internal consistency. This algorithm piggybacks abort and crash information on the normal messages between nodes. We consider a simpler algorithm that only handles orphans arising from explicit aborts. We describe the simplified orphan detection algorithm at various levels of abstraction, using an algebraic model convenient for describing asynchronous systems. The highest-level model is specified in terms of a (virtual) global state. At this level of abstraction we require that the states generated by the model satisfy view-serialability. Lower-level models progressively localize the description of the algorithm's operation, and the lowest level of abstraction presents a fully distributed model of the (simplified) orphan detection scheme.

TR-287
Bui, T.N.
ON BISECTING RANDOM GRAPHS
Pages: 45 S.M. Thesis/March 1983 $4.70
Keywords: random graphs, bisection size, probabilistic bounds
Abstract: A bisection of a graph with an even number of vertices is a partition of the vertex set into two disjoint sets of equal size. Given a bisection, the number of edges having one end in each of the two subsets of the bisection is called the size of the bisection. The bisection size of the graph is the minimum size of all possible bisections of the graph. Given a graph with an even number of vertices and a positive integer, the graph bisection problem is the problem of determining if the bisection size of the graph is less than the given number. The graph bisection problem is known to be NP-hard.

In this thesis, we give probabilistic lower bounds and upper bounds for the bisection size of random graphs, graphs in which an edge appears between any two vertices with a certain fixed probability, say p, independent of all other edges. In particular, we show that, with probability 1, the bisection size of random graphs on $2n$ vertices is greater than or equal to $n^2p\text{-}0(n^{3/2}\sqrt{p(1\text{-}p)})$ and is less than $n^2p\text{-}0(n\sqrt{p(1\text{-}p)})$. Upper bound and lower bound on the bisection size are given in the case p is a function of n, specifically when $p = p(n) = c/n$. We also consider some heuristics for solving the graph bisection problem.

TR-288
Landau, S.E.
ON COMPUTING GALOIS GROUPS AND ITS APPLICATION TO SOLVABILITY BY RADICALS
Pages: 76 Ph.D. Dissertation/March 1983 $5.60
Keywords: Galois groups, solvability by radicals, polynomial-time, mathematics
Abstract: This thesis presents a polynomial time algorithm for the basic question of Galois theory, checking the solvability by radicals of a monic irreducible polynomial over the integers. It also presents polynomial time algorithms for factoring polynomials over algebraic number fields, for computing blocks of imprimitivity of roots of a polynomial under the transitive action of number fields. (In all of these algorithms it is assumed that the algebraic number field is given by a primitive element which generates it over the rationals, and that the polynomial in question is monic, with coefficients in the integers.) We also show how to express a root in radicals in terms of a straightline program in polynomial time.

The techniques used include methods from computational complexity and approaches from the theory of finite permutation groups. The results presented here rely on the recent work of Lenstra, Lenstra, and Lovasz, in which a polynomial time algorithm for factoring polynomials over the integers is presented.

Many questions remain. Our divide-and-conquer approach answers the question of solvability without revealing the nature of the group in question; we do not even learn its order. We suggest this as one of the many open problems that remain to be tackled.

TR-289
Sirbu, M., Schoichet, S.R., Kunin, J.S., Hammer, M.M., Sutherland, J.B., Zarmer, C.L.
OFFICE ANALYSIS: METHODOLOGY AND CASE STUDIES
Pages: 95 March 1983 $6.20
Keywords: office analysis, systems analysis, office automation, automated office systems, office model
Abstract: The Office Analysis Methodology (OAM) is a structured methodology for understanding the current operations of an office. OAM provides guidance in interviewing techniques and approaches to establishing a positive atmosphere for possible office automation efforts. It is designed to be easy to learn so that people with experience in office work but little experience in analysis can easily perform a study. OAM makes use of an office work model to ease the data gathering process. The product of an OAM study is a document, organized according to a standard format, describing the current operations of the office.

This report contains the original memo describing OAM, along with the OAM descriptions of five MIT offices.

TR-290
Sutherland, J.B.
AN OFFICE ANALYSIS AND DIAGNOSIS METHODOLOGY
Pages: 156 S.M. THESIS/March 1983 $8.00
Keywords: office automation, automated office systems, office analysis, understanding office work, office model
Abstract: With the advent of computer technology designed for use in the office, office analysis, or the process of understanding office work for the purposes of introducing technology, has become increasingly important. The Office Analysis and Diagnosis Methodology (ODAM) is a tool to help the analyst gather the data required to decide how, and whether, to introduce office automation technology into a particular office. OADM is best suited for studying semi-structured offices, rather than pure processing operations or specal projects. OADM is used to perform a detailed study of a single office and is not designed for use in determining the general automation needs of a large organization.

Initial experience with OADM's ideological parent the Office Analysis Methodology (OAM) suggested areas for change and OADM is designed to overcome the perceived weaknesses of OAM. An evaluation of OAM presented in this thesis confirms its limitations. OADM has not been evaluated directly, however, the evaluation of OAM shows that the most significant differences between OAM and OADM should make OADM a more useful methodology.

Chapters 2, 3 and the appendices of this thesis form a complete manual for the use of OADM, but do not provide a description of the office model on which OADM is based.

TR-291
Pinter, R.Y.
THE IMPACT OF LAYER ASSIGNMENT METHODS ON LAYOUT ALGORITHMS FOR INTEGRATED CIRCUITS
Pages: 144 Ph.D. Dissertation/August 1982 $7.65
Keywords: analysis of algorithms, computational geometry, design automation, graph theory, layout of integrated circuits, placement and routing, river routing, VLSI
Abstract: Programs for integrated circuit layout at the module assembly level are typically decomposed into two phases - placement and routing. In this thesis we investigate a third phase which is often implicitly assumed -layer assignment. This thesis studies how layer assignment methodologies interact with placement and routing.

A simple layer assignment methodology, which is also commonly occurring, is *river routing*, where all wires can be routed in a single layer. We give concise necessary and sufficient conditions for a channel to be river routable, and based on these conditions, give a linear-time algorithm to find the optimal placement of modules across a channel. We also show that determining whether wires can be river routed in an arbitrary polygon can be determined quickly. In addition, we give NP-completeness results for several other planar routing problems.

In addition, we define new characteristics of channel routing: *monotonicity* and *jogging*, and investigate their relation to minimizing the channel's width. In both the *Manhattan* and the *knock-knee* two-layer routability of certain configurations in **T**- and **X**-shaped channels can be checked in linear time.

Finally, we discuss optimal layer assignment as an "after the fact" consideration, when wiring is specified without layer designation. We give an $0(n^{2.5})$ algorithm for minimizing the number contacts required for two-layer realizations.

A glossary of layout problems is provided as an appendix.

TR-292
Dornbrook, M., Blank, M.
THE MDL PROGRAMMING LANGUAGE PRIMER
Pages: 125 June 1980 $7.10
Keywords: programming languages, MDL, Muddle
Abstract: Over the years the original MDL (pronounced "Muddle") Primer by Greg Pfister became more and more a reference manual and less a Primer from which a novice could learn the language. Some of the text of the original has been re-used in this document, but much has been eliminated, changed, or re-ordered, and a reasonable amount of new material has been added. In particular, a number of figures and many more examples have been added to make some of the more difficult concepts easier to understand.

This Primer is intended as an introduction to MDL. After assimilating the information contained herein, you should be able to write very good programs. However, for any individual topic in the MDL Primer there is likely to be more information available in *The MDL Programming Language* and *The MDL Programming Environment*, and there are many topics in these documents which are not addressed in the Primer. Anyone who plans to do any serious work with MDL should read these documents.

One of the difficulties in writing a Primer is to make it useful to those who don't know anything at all about programming without boring those who know a lot of the basics. Hopefully those at both extremes will find this to be easy to read. If you are a complete novice, however, there may be some unfamiliar references and some material which doesn't make sense in your first reading.

TR-293
Galley, S.W., Pfister, G.
THE MDL PROGRAMMING LANGUAGE
Pages: 276 May 1979 $11.60
Keywords: MDL, Muddle, programming languages
Abstract: The MDL programming language began existence in late 1970 (under the name Muddle) as a successor to LISP, a candidate vehicle for the Dynamic Modeling System, and a possible base for implementation of Planner. The original design goals included an interactive integrated environment for programming, debugging, loading, and editing: ease in learning and use: facilities for structured, modular, shared programs: extensibility of syntax, data types and operators: data-type checking for debugging and optional data-type declarations for compiled efficiency: associative storage, corouting, and graphics. Along the way to reaching those goals, it developed flexible input/output (including the ARPA Network), and flexible interrupt and signal handling. It now serves as a base for software prototyping, research, development, education, and implementation of the majority of programs at MIT-DMS: a library of sharable modules, a coherent user interface, special research projects, autonomous daemons, etc.

This document was originally intended to be a simple low-level introduction to MDL. It has, however, acquired a case of elephantiasis and now amounts to a discursive description of the whole interpreter, as realized in MDL release numbers 55 (ITS version) and 105 (Tenex and Tops-20 versions). (Significant changes from the previous edition are marked in the margin.) A low-level introduction nay still be had by restricting one's attention to specially-marked sections only. The scope of the document is confined as much as possible to the interpreter itself. Other adjuncts (compiler, assembler, pre-loaded user programs, library) are mentioned as little as possible, despite their value in promoting the language seen by a user from "basic survival" to "comfortable living". Indeed, MDL could not fulfill the above design goals without the compiler, assembler, structure editor, control-stack printer, context printer, pretty-printer, dynamic loader, and library system -- all of which are not part of the interpreter but programs written in MDL and symbiotic with one another. Further information on these adjuncts can be found in Lebling's document.

TR-294
Lebling, P.D.
THE MDL PROGRAMMING ENVIRONMENT
Pages: 137 May 1980 $7.45
Keywords: MDL, programming languages, Muddle
Abstract: The MDL language is described in "The MDL Programming Language", but in addition to the language itself, there is a rich and varied collection of software written in the language which facilitates the writing of programs and systems of programs in MDL. The information describing this programming environment has been contained in various documents, some out of print or out of date, and in supplemental disk files describing changes and additions. Some of the packages of functions used deal with MDL code have never been formally documented. This manual brings together some of that scattered documentation.

The document's purpose is to flesh out the description of the language in "The MDL Programming Language", giving a fuller description of the program writing and debugging aids available to MDL users, to describe the methods for producing code usable by others, to describe the MDL compiler and the many other techniques for producing and speeding up MDL object code.

The imagined reader of this document is someone who has read "The MDL Programming Language", and now proposes to write programs in MDL, possibly even very large programs. MDL packages that he would find useful in the *process* of doing so are documented here: editors, debuggers, etc. Packages that he might wish to use *within* his program are not included: data-management systems, command interpreters, etc.

This document is of necessity highly self-referent, as many of the components of the MDL programming refer to each other and adhere to the same conventions. Additionally, this document assumes that the reader is familiar with the language itself (at least to some degree) and with the ITS, TENEX, or TOPS-20 operating systems.

TR-295
Pitman, K.M.
THE REVISED MACLISP MANUAL
Pages: 325 June 1983 $13.10
Keywords: MACLISP, artificial intelligence, LISP, list structure, symbol manipulation
Abstract: MACLISP is a dialect of Lisp developed at M.I.T.'s Project MAC (now the MIT Laboratory for Computer Science) and the MIT Artificial Intelligence Laboratory for use in artificial intelligence research and related fields. Maclisp is descended from Lisp 1.5, and many recent important dialects (for example Lisp Machine Lisp and NIL) have evolved from Maclisp.

David Moon's original document on Maclisp, *The Maclisp Reference Manual* (alias the *Moonual*) provided in-depth coverage of a number of areas of the Maclisp world. Some parts of that document, however, were never completed (most notably a description of Maclisp's I/O system); other parts are no longer accurate due to changes that have occurred in the language over time.

This manual includes some introductory information about Lisp, but is not intended as tutorial. It is intended primarily as a reference manual; particularly, it comes in response to user's please for more up-to-date documentation. Much text has been borrowed directly from the *Moonual*, but there has been a shift in emphasis. While the *Moonual* went into greater depth on some issues, this manual attempts to offer more in the way of examples and style notes. Also, since Moon had worked on the Multics implementation, the *Moonual* offered more detail about compatability between ITS and Multics Maclisp. While it is hoped that Multics users will still find the information contained herein to be useful, this manual focuses more on the ITS and TOPS-20 implementations since those were the implementations most familiar to the author.

TR-296
Church, K.W.

Pages: 175 Ph.D. dissertation/June 1983 $8.60
Keywords: speech recognition, parsing, grammars, allophonic variation, phonotactics, constraints, lexical retrieval, syllable
Abstract: It is well-known that phonemes have different acoustic/phonetic realizations depending on the context. Thus, for example, the phoneme /t/ is typically realized with a heavily aspirated strong burst at the beginning of a syllable, as in the word *Tom*, but without a burst at the end of a syllable, in a word like *cat*. It is common practice in speech research to distinguish between two types of acoustic/phonetic features: (a) those that vary a great deal with context (e.g., aspiration) and (b) those that are relatively invariant to context (e.g., place, manner, voicing). In the past, the emphasis has been

on invariants; allophonic variation is traditionally seen as problematic for recognition. However, it is well-known that allophonic contrasts can be distinctive, as illustrated by famous minimal pairs such as *a tease* vs. *at ease*, *night rate* vs. *nitrate*, *great wine* vs. *gray twine*, etc. This sort of evidence suggests that allophonic variation provides a rich source of constraint on syllable structure and word stress. The recognizer discussed (but only partly implemented) in my thesis is designed to exploit allophonic and phonotactic cues by parsing the input utterance into syllables and other suprasegmental constituents using phrase-structure parsing techniques.

TR-297
Mok, A.K.
FUNDAMENTAL DESIGN PROBLEMS OF DISTRIBUTED SYSTEMS FOR THE HARD-REAL-TIME ENVIRONMENT
Pages: 150 Ph.D. dissertation/June 1983 $7.85
Keywords: hard real-time systems, real-time systems, real-time scheduling, distributed systems, software engineering, design automation, embedded systems
Abstract: Software designed to function in a hard-real-time environment where strict timing constraints must be met often entails implicit assumptions about a programming language and the underlying system which supports it. Programs which are logically correct, i.e., they implement the intended algorithms, may not function correctly if their assumed timing characteristics of the software or if the expressible timing characteristics cannot be verified before run time. For distributed systems in particular, the software must be tailored to a myraid of implementation parameters, e.g., communication bandwidth, thus rendering subsequent modifications hazardous.

Our research investigates the basic problems in automating the design and maintenance of hard real-time software. After examining the limitations of the traditional approach to real-time software design via process-based models, we shall provide a graph-based computation model which is more suitable for expressing the computational requirments of the real-time environment. This model is an extension of CONSORT (Control Structure Optimized for Real-Time), an experimental software design system which has been implemented to generate process control application programs from block diagram schemata. While our graph-based model is abstract, it can serve as a useful intermediate representation between textual requirments specifications and target application programs. Using the graph-based model, the complexity of the relevant resource allocation problems for meeting stringent timing constraints is investigated.

TR-298
Krugler, K.
VIDEO GAMES AND COMPUTER AIDED INSTRUCTION
Pages: 45 June 1983 $4.70
Keywords: CAI, video games, IBM PC
Abstract: This document will briefly outline the evolution of video games, discuss current video game theory, and describe a program to teach typing on the IBM Personal Computer.

TECHNICAL MEMORANDA ABSTRACTS

TM-10
Jackson, J.N.
INTERACTIVE DESIGN COORDINATION FOR THE BUILDING INDUSTRY
Pages: 32 June 1970 $3.00
Keywords: building design, building construction, ICES system, structure analysis
Abstract: The problem of effective communication in the process of building design and construction is widely recognized. The involvement of several design disciplines combined with the tendency for designers to work in distinct offices results in little capacity for them to investigate the influence of their design decisions on other design areas.

One of the responses to the need for effective interaction in the use of computers for a design project is the supersystem concept proposed for ICES, the Integrated Civil Engineering System. The supersystem is defined as the cooperative effort on the part of the designers of several problem oriented computer capabilities to implement project oriented capabilities by allowing each of their problem oriented subsystems to reference a single file of project data. The supersystem would allow design interaction by having each of the problem oriented computer subsystems reference a single file of information specifying the project.

Future work in the application of computers to interactive and project oriented design in the building industry will have to concentrate on the file structure to be used in the implementation of a computer building design supersystem. {AD 708-400}

TM-11
Ward, P.W.
DESCRIPTION AND FLOW CHART OF THE PDP-7/9 COMMUNICATIONS PACKAGE
Pages: 60 July 1970 $3.00
Keywords: computer communications, data links
Abstract: The PDP-7/9 Communications Package was written to provide data transfers between the buffer controller PDP-7 or PDP-9 of an ESL Display Console and a host computer via a 50 kilobit serial telephone link using Bell System Type 303 Dataphones. This memorandum describes the package programs and calling procedures, and includes detailed flow diagrams. {AD 711-379}

TM-12
Graham, R.M
FILE MANAGEMENT AND RELATED TOPICS
Pages: 53 September 1970 $3.00
Keywords: file systems, virtual memory, paging, segmented, time-sharing, memory management, multi-level file storage, file sharing, file protection
Abstract: This paper traces the evolution of a segment based file system. The final system is typical of the virtual memory systems found in large general purpose time-sharing systems. The contents of the file system is a collection of symbolically named segments organized in a hierarchical structure. The user directly references segments in the file system. All movement of information between the different levels of physical memory is done automatically by the system using paging. Complete privacy of user information is guaranteed, although controlled sharing is possible. The system includes file backup facilities to protect users from information loss due to system failure. {AD 712-068}

TM-13
Graham, R.M.
USE OF HIGH LEVEL LANGUAGES FOR SYSTEMS PROGRAMMING
Pages: 21 September 1970 $3.00
Keywords: programming languages, systems programming, project management, system design, system implementation, system performance analysis, high levellanguages, system modeling
Abstract: The basic problems in the design and implementation of large software systems are reviewed. Using a high level language, such as PL/1, to implement a large software system has many advantages. Several of the major advantages and how they contribute to the solution of the major implementation problems are discussed. It is pointed out that none of the high level languages existing today help in solving the problem of performance prediction. It is then postulated that a language designed specifically for software design and implementation would not only be a major factor in the solution of the basic problems of software design and implementation previously discussed, but it would also make it possible to automatically predict the performance of the software being designed. Some properties of such a language are explored. A direction for obtaining the performance measure through the use of analysis and simulation is explored. {AD 711-965}

TM-14
Vogt, C.M.
SUSPENSION OF PROCESSES IN A MULTIPROCESSING COMPUTER SYSTEM
Pages: 79 S.M. Thesis/September 1970 $3.00
Keywords: supervisory systems, multiprocessing, time-sharing, interrupts, operating systems, swapping
Abstract: This document defines the notion of a suspension capability and sets down the requirements such a capability makes on a computing system. A simple model of a computing system enables a more detailed statement of those requirements. A cursory investigation of the Multics computing system shows why it is difficult to implement a suspension capability in multiprocessing systems with direct sharing, and, in particular, in Multics. {AD 713-989}

TM-15
Zilles, S.N.
AN EXPANSION OF THE DATA STRUCTURING CAPABILITIES OF PAL
Pages: 201 S.M. Thesis/October 1970 $3.00
Keywords: programming languages, extensible programming languages, PAL, data structures
Abstract: PAL is a language designed for use as a tool to help teach programming linguistics. PAL is extended to include additional facilities for structuring data.

The structure definitions of Landin are incorporated into the PAL syntax. The data structures are represented by functions defined on a set of symbolic component selectors. A type system based on unrestricted predicate functions is introduced to provide strong representations of the data structures.

The new language features are formally defined by appropriate modifications to the existing formal definition of PAL. The flexibility and power of the extensions is illustrated in a series of examples. Limitations, alternatives and possible extensions are discussed. {AD 720-761}

TM-16
Bruere-Dawson, G.
PSEUDO-RANDOM SEQUENCES
Pages: 54 S.M. Thesis/October 1970 $3.00
Keywords: recursive functions, Church random sequences, sequential tests, probability laws, descriptive complexity, Kleene hierarchy
Abstract: Three definitions of random binary sequences are presented The consistency of those definitions with the laws of probability theory, and the inclusion relationship of the three sets of random sequences, are investigated.

These sequences, considered as characteristic functions of sets, are then placed in the Kleene arithmetical hierarchy. Some restrictions on these definitions, using Blum's complexity theory, lead to the definition of pseudo-random sequences, which can be generated effectively. {AD 713-852}

TM-17
Goodman, L.I.
COMPLEXITY MEASURES FOR PROGRAMMING LANGUAGES
Pages: 85 S.M. Thesis/September 1971 $3.00
Keywords: computational complexity, program resource usage, complexity measures, programming languages, program equations, program equivalence relations
Abstract: A theory of complexity is developed for algorithms implemented in typical programming languages. The complexity of a program may be

interpreted in many different ways; a method for measuring a specific type of complexity is a complexity measure -- some function of the amount of a particular resource used by a program in processing an input. Typical resources would be execution time, core, I/O devices, and channels.

Any resource whose use is independent of previous and future usage can be handled by the theory. This condition includes time but excludes space complexity. For specific measure, the complexity of an arbitrary program with a particular input. Because this method gives little information about the general complexity behavior of the program, another approach is developed.

This new approach analyzes the complexity of a program with respect to a valid set of inputs -- a finite set of legitimate, halting inputs. A program equation is developed to make the transformations undergone by the inputs more explicit. Using the equation, the input set is partitioned into classes of constant complexity. The classes are used to compute maximum, minimum, and expected complexities of the program on the input set.

Several equivalence relations are defined, relating different programs by their complexity. Complexity is also discussed in terms of concatenation and functional equivalence of programs. {AD 729-011}

TM-18
Miller, P.L.
AUTOMATIC CODE-GENERATION FROM AN OBJECT-MACHINE DESCRIPTION
Pages: 74 October 1970 $3.00
Keywords: compilers, translator, macroprocessor, code-generation, automatic code generation
Abstract: This report outlines the basic elements of a macro code-generating system, and develops an informal machine-independent model of a code generator. Then the report discusses how an implementation of this model could be set up to generate code for a particular machine from machine-dependent information given in descriptive form.{AD 713-853}

TM-19
Fenichel, R.R.
A NEW LIST-TRACING ALGORITHM
Pages: 26 October 1970 $3.00
Keywords: list-processing, garbage collection, storage allocation
Abstract: List-processing systems have each allowed use of only a single size and configuration of list cell. This paper describes a system which allows use of arbitrarily many different sizes and configurations of list cell, possibly not specified until run time. {AD 714-522}

TM-20
Jones, T.L.
A COMPUTER MODEL OF SIMPLE FORMS OF LEARNING
Pages: 146 Ph. D. Dissertation/January 1971 $3.00
Keywords: computer learning, learning models, artificial intelligence
Abstract: A basic unsolved problem in science is that of understanding learning, the process by which people and machines use their experience in a situation to guide future actions in similar situations. This thesis presents an approach to the learning problem and a learning-oriented approach to the artificial intelligence problem. These approaches are illustrated in a computer program called INSIMI, which models simple forms of learning analogous to the learning of a human infant during the first few weeks of his life, such as learning to suck the thumb and learning to perform elementary hand-eye coordination.

The program operates by discovering cause-effect relationships and arranging them in a goal tree. For example, if A causes B, and the program wants B, it will set up a as a subgoal, working backward along the chain of causation until it reaches a subgoal which can be reached directly; i.e., a muscle pull

The work is discussed in relation to fundamental scientific issues, and proposals are made for future research. {AD 720-337}

TM-21
Goldstein, R.C.
THE SUBSTANTIVE USE OF COMPUTERS FOR INTELLECTUAL ACTIVITIES
Pages: 18 April 1971 $3.00
Keywords: man-computer interaction, interactive systems, management information systems
Abstract: This paper discusses an on-going research project aimed at developing computer facilities capable of providing substantive aid to a human decision maker concerned with complex, unstructured problems. The rationale for such systems is discussed, followed by an outline of the approach used. Some results of preliminary experiments are also discussed, as well as plans for future activities. {AD 721-618}

TM-22
Wells, D.
TRANSMISSION OF INFORMATION BETWEEN A MAN-MACHINE DECISION SYSTEM AND ITS ENVIRONMENT
Pages: 11 April 1971 $3.00
Keywords: canonical form, functional modularity, man-machine decision system, external interfaces, information systems
Abstract: This paper describes the structure of the external communication facilities of a highly-interactive information system designed to assist a user in making non-trivial decisions. The report also examines the usefulness of functional modularity and specification of canonical form interfaces as an aid in the comprehension of the interaction of the various modules of a complex system. The areas of external communication defined and examined are: data collection, process management, report generation, and facilities management. {AD 722-837}

TM-23
Strnad, A.J.
THE RELATIONAL APPROACH TO THE MANAGEMENT OF DATA BASES
Pages: 26 April 1971 $3.00
Keywords: data management system, data manipulation, management information systems, information retrieval, access control, data base management, set-theoretical data structures, data organization, data structure
Abstract: This paper is concerned with the design and implementation of a relational system for management of Large Data Bases [LDB] at M.I.T. Project MAC.

We have determined the following six major requirements for the management of LDB in a dynamically varying environment, such as an Interactive Management System: high degree of flexibility, data independence, ability to operate on different data structures, access path independent of data structure, access control below the file level, uniform retrieval time.

We take the view that information we might store in our LDB consists of sets of data elements and sets of relations among data elements. The basic set theoretical operations are used for manipulating and operating upon sets. {AD 721-619}

TM-24
Goldstein, R.C., Strnad, A.J.
THE MACAIMS DATA MANAGEMENT SYSTEM
Pages: 29 April 1971 $3.00
Keywords: data management system, information retrieval, data structure,data manipulation, access control, data organization, management information systems, data base management, set-theoretical data structures
Abstract: This paper describes the MACAIMS Data Management System (MADAM). It begins with a brief discussion of the overall goals of the project, its operating environment (Multics), and its data management requirements.

The MADAM system is based on a relational model of data, and employs set-theoretic primitive operations for manipulating data. Theasic philosophy of the system and some issues involved in its implementation are described. {AD 721-620}

TM-25
Goldstein, R.C.
HELPING PEOPLE THINK
Pages: 22 April 1971 $3.00
Keywords: man-computer interaction, interactive systems, management information systems
Abstract: This paper describes some results of research in interactive problem solving and decision making. The requirements of a computer system for aiding a human decision maker are described with particular reference to the nature of the man-computer interface.

Some examples of interactive programs developed during the course of this research are also included. {AD 721-998}

TM-26
Iaezolla, G.G.
MODELING AND DECOMPOSITION OF INFORMATION SYSTEMS FOR PERFORMANCE EVALUATION
Pages: 54 June 1971 $3.00
Keywords: information systems, modeling, computer systems, computer networks, decomposition, performance
Abstract: The problem of evaluation computer systems performance is the leading motivation of this work, whose aim is to propose premises for configuration-independent evaluation criteria. The existing works on the subject are first reviewed and are found generally bound to particular system configurations and based upon parameter choices that vary from author to author. This increases the necessity to establish common, analytic, basic metrics and equations prior to any further discussion of performance. To this purpose, two topics are introduced: modeling of information systems and decomposition of information systems. The first provides means to quantitatively characterize how the performance of an operating unit (CPU, memory, etc.) inside the system differs from that when set apart from it, it also permits the study of the optimal way of putting together given set of units to form a computer system or a computer network. The second argument contains a case study, where a method of analysis is introduced that is based on the decomposition of the system or the network into hierarchical chains of units.

This approach led to the adoption of an unfamiliar symbolism, which proved to be a fruitful "complication" as soon as we obtained the first results. {AD 733-965}

TM-27
Bagchi, A.
ECONOMY OF DESCRIPTIONS AND MINIMAL INDICES
Pages: 36 January 1972 $3.00
Keywords: minimal indices, theory of computation
Abstract: In part One sets of minimal indices M_s and M are defined. It is shown that M_s amd M are immune and that $M\ {}^{7}{}_{T}\ \Phi''$. M_s join $K^{7}\ {}_{T}\ \Phi''$. Subsets of M called M_N and M_F are defined and it is proved that $M_F\ {}^{7}{}_{T}\ \Phi'$ and that $M_N\ {}^{7}{}_{T} K \cap M\ {}^{7}{}_{T} K \cap M\ {}^{7}{}_{T}\ \Phi''$. M_s is relativized with respect to a set A of integers, and for any two sets A and B of integers such that $A'' \leq_T B'$ and any total function $g \leq_T B''$ and a size function $s \leq_T A$ the following set C is shown to be nonempty

$$C\ {}^{7}\ \{y \mid \exists x[W_x^A = W_y^B \text{ and } x \in M_s^A \text{ and } s(4) > g(y)]\}$$

C however is empty for some total functions $g \leq_T B'''$. Various special cases are considered, e.g. W_y^B in the definition of C is restricted to be finite or a singleton.

TM-28
Wong, R.
CONSTRUCTION HEURISTICS FOR GEOMETRY AND A VECTOR ALGEBRA REPRESENTATION OF GEOMETRY
Pages: 100 June 1972 $3.00
Keywords: mathematics, heuristics, vector algebra, geometry
Abstract: Heuristics for generating constructions to help solve high school geometry problems are given. Many examples of the use of these heuristics are given. A method of translating geometry problems into vector algebra problems is discussed. The solution of these vector algebra geometry problems is analyzed. The use of algebraic constructions to help solve these vector problems is also discussed. {AD 743-487}

TM-29
Hossley, R., Rackoff, C.
THE EMPTINESS PROBLEM FOR AUTOMATA ON INFINITE TREES
Pages: 16 June 1972 $3.00
Keywords: finite tree automata, infinite tree automata, emptiness problem
Abstract: The purpose of this paper is to give an alternative proof to the decidability of the emptiness problem for tree automata, as shown in Rabin [4]. The proof reduces the emptiness problem for automata on infinite trees to that for automata on finite , by showing that any automata definable set of infinite trees must contain a finitely-generable tree. {AD 747-250}

TM-30
Mc Cray, W.M.
SIM360: A S/360 SIMULATOR
Pages: 117 S.B. Thesis/October 1972 $3.00
Keywords: computer systems, operating systems
Abstract: Modern, large-scale computer systems typically operate under the control of an operating system or executive program, and reserve for the exclusion use of the operating system a set of privileged instructions, which the normal users may not issue. This very necessary arrangement produces a problem of equipment availability for those who wish to develop or investigate operating systems programs, because such programs cannot be run as normal user jobs under an executive program

This thesis describes SIM360, a detailed simulator of a representative IBM S/360 computer, which was written to run student programs, programs assigned as machine problems for a course in operating systems. The simulator allows programs to issue all of the privileged instructions of the S/360, and thus provides a readily available tool for the study of operating systems programs. {AD 749-365}

TM-31
Bonneau, R.J.
A CLASS OF FINITE COMPUTATION STRUCTURES SUPPORTING THE FAST FOURIER TRANSFORM
Pages: 15 March 1973 $3.00
Keywords: Fourier transforms, mathematics, fast Fourier transforms
Abstract: The Fast Fourier Transform (FFT) and modular arithmetic are two distinct techniques which recently have been employed to increase the efficiency of numerous algorithms in the area of symbolic and algebraic manipulation. Motivated by work done on fast large integer multiplication by Schonhage and Strassen [11] and by Knuth [7], this paper analyzes the question of when these two techniques can be utilized concurrently. The desirability of the convolution property of the FFT suggests a practical definition for the support of an FFT, while a generalization of the modular rings of integers motivates a reasonable definition of a finite computation structure. A Finite Computation Structure is defined to be a commutative ring with unity, and of finite, non-zero characteristic. This report first completely characterizes the modular rings of integers which support the FFT by considering the prime factorization of the modulus. This characterization is then extended to provide the following result: Theorem: Let R be a finite computation structure of characteristic m. Then R will support a K-point FFT if K divides p-1 for each prime p dividing m. The paper then concludes with examples of the application of this result to the problems of computing products and powers of symbolic multivariate polynomials. {AD 757-787}

TM-32
Moll, R.
AN OPERATOR EMBEDDING THEOREM FOR COMPLEXITY CLASSES OF RECURSIVE FUNCTIONS
Pages: 11 May 1973 $3.00
Keywords: recursive functions, operators, complexity
Abstract: Let ***F***(t) be the set of functions computable by some machine using no more than t(x) machine steps on all but finitely many arguments x. If we order the ***F***-classes under set inclusion as t varies over the recursive functions, then it is natural to ask how rich a structure is obtained. We show that this structure is very rich indeed. If R is any countable partial order and **F** is any total effective operator, then we show that there is a recursively enumerable sequence of recursive machine running times $<\Phi_{s(k)}>_{k \in \mathbf{N}}$ such that if jRk, then $\Phi(F(\Phi_{s(j)})) \cap \neq \boldsymbol{F}(\Phi_{s(k)})$, and if j and k are incomparable, then $\mathbf{F}(\Phi_{s(j)}) < \Phi_{s(k)}$ on infinitely many arguments, and $\mathbf{F}(\Phi_{s(k)}) < \Phi_{s(j)}$ on infinitely many arguments.

An interesting feature of our proof is that we avoid appealing explicitly to the continuity of total effective operators; indeed our proof follows directly from a single appeal to the recursion theorem.

Several investigators have considered this and related problems, and in Section 4 we briefly summarize these investigations and compare them to our own. {AD 759-999}

TM-33
Ferrante, J., Rackoff, C.
A DECISION PROCEDURE FOR THE FIRST ORDER THEORY OF REAL ADDITION WITH ORDER
Pages: 16 May 1973 $3.00
Keywords: real addition, mathematics
Abstract: Consider the first order theory of the real numbers with the predicates + (plus) and < (less than). Let S be the set of true sentences. We first present an elimination of quantifiers decision procedure for S, and then analyze it to show that it takes at most time $2^{2^{cn}}$, c a constant, to show decide sentences of length n.

Looking more closely at this procedure, we arrive at a second procedure by showing that a given sentence doesn't change in truth value when each of the quantifiers is limited to range over an appropriately chosen finite set of rationals. This fact leads to a decision procedure for S which takes space 2^{cn}. We also remark that our methods lead to a decision procedure for Presburger arithmetic which operates in *space* $2^{2^{cn}}$.

These upper bounds should be compared with the results of Fischer and Rabin (Proceedings of A.M.S. Symposium on Complexity of Real Computation Processes, to appear) that for some constant c, time 2^{cn} for real addition, and time $2^{2^{cn}}$ for Presburger arithmetic, is required to decide some sentences of length n for infinitely many n. {AD 760-000}

TM-34
Bonneau, R.J.
POLYNOMIAL EXPONENTIATION: THE FAST FOURIER TRANSFORM REVISITED
Pages: 35 June 1973 $3.00
Keywords: mathematics, Fourier transorms, fast Fourier transform, polynomial exponentiation
Abstract: The Fast Fourier Transform (FFT) is a method proposed for the computation of powers of symbolic multivariate polynomials over the integers. Despite its acknowledged superiority in as inefficient for practical systems. This report presents concrete evidence to support the claim that the FFT method is a highly efficient algorithm for the practical computation of the powers of polynomials.

The report proceeds by defining the Discrete Fourier Transform (DFT) and its inverse, and detailing the relationship between the DFT and the FFT. The convolution property of the FFT is then stated, along with its applications to univariate polynomial multiplication and exponentiation. These applications are then extended to include multivariate polynomials by considering the computation structures in which the FFT may be performed. Several problems concerned with the implementation of the FFT are discussed and solutions are given for the actual system implementation. Finally, conclusions on the efficiency of the FFT algorithm are drawn from timing results obtained form extensive testing of the FFT and other proposed methods. {PB 221-742}

TM-35
Bonneau, R. J.
AN INTERACTIVE IMPLEMENTATION OF THE TODD-COXETER ALGORITHM
Pages: 24 December 1973 $3.00
Keywords: Todd-Coxeter algorithm
Abstract: The Todd-Coxeter algorithm provides a systematic approach to the enumeration of cosets of a finitely presented group. This memo describes an interactive implementation of algorithm, including a manual on its use, examples, and methods of accessing the program. Applications of this algorithm are also discussed. {AD 770-565}

TM-36
Geiger, S. P.
A USER'S GUIDE TO THE MACRO CONTROL LANGUAGE
Pages: 38 December 1973 $3.00
Keywords: macro control language
Abstract: The purpose of this guide is to explain the syntax and semantics of the statements in the Macro Control Language. The guide assumes that the reader is familiar with the assembly language for the PDP-11. The Macro Control Language is the base of a new language approach which combines the advantages of compilation from a higher-level language with the automatic scheduling of a pre-programmed real-time system. {AD 771-435}

TM-37
Schonage, A.
REAL-TIME SIMULATION OF MULTIDIMENSIONAL TURING MACHINES BY STORAGE MODIFICATION MACHINES
Pages: 7 December 1973 $3.00
Keywords: Turing machine, storage modification machines
Abstract: In [1] the author introduced a new machine model, now called the Storage Modification Machine (SMM). It was claimed, but not proved, that SMM's can simulate all sorts of Turing machines — those with multidimensional worktapes in particular — in real time.

In the following sections we describe a real-time simulation technique for keeping track of the movements of a Turing machine read-write head in the plane in such a way that repeated visits to a square are properly recognized. This construction shows how to overcome the main difficulty in real-time simulation by SMM's of multidimensional storage devices, a difficulty which typically arises in the two-dimensional case. Furthermore, this will reinforce the intuition that such multidimensional worktapes do not possess any latent computational power. {PB 226-103/AS}

TM-38
Meyer, A. R.
WEAK MONADIC SECOND ORDER THEORY OF SUCCESSOR IS NOT ELEMENTARY- RECURSIVE
Pages: 24 $3.00
Keywords: logics of programs, monadic second order theory
Abstract: Let L_{SIS} be the set of formulas expressible in a weak monadic second order logic using only the predicates $[x=y+1]$ and $[x \in X]$. Buchi and Elgot have shown that the truth of sentences in L_{SIS} (under the standard interpretation $<N$, successor$>$ with second order variables interpreted as ranging over finite sets) is decidable. We refer to the true sentences in L_{SIS} as WSIS. We shall prove that WSIS is not elementary-recursive in the sense of KalMarch In fact, we claim a stronger result. {PB 226-514/AS}

TM-39
Meyer, A.R.
DISCRETE COMPUTATION: THEORY AND OPEN PROBLEMS
Pages: 35 January 1974 $3.00
Keywords: discrete computation, theory of computation
Abstract: Notes for the lectures by Prof. Meyer: Preceptorial Introduction to Computer Science for Mathematicians, The American Mathematical Society. {PB 226-836/AS}

TM-40
Paterson, M.S, Fischer, M.J., Meyer, A.R.
AN IMPROVED OVERLAP ARGUMENT FOR ON-LINE MULTIPLICATION
Pages: 28 January 1974 $3.00
Keywords: mathematics, Turing machine, overlap argument
Abstract: A lower bound of cNlogN is proved for the mean time complexity of an on-line multitape Turing Machine performing the multiplication of N-digit binary integers. For a more general class of machines which includes some models of random-access machines, the corresponding bound is cNlogN/loglogN. These bounds compare favorably with known upper bounds of the form $cN(\log N)^k$, and for some classes the upper and lower bounds coincide. The proofs are based on the "overlap" argument due to Cook and Aanderaa. {AD 773-137}

TM-41
Fischer, M.J., Paterson, M.S.
STRING-MATCHING AND OTHER PRODUCTS
Pages: 21 January 1974 $3.00
Keywords: string-matching, Turing machine
Abstract: The string-matching problem considered here is to find all occurrences of a given pattern as a substring of another longer string. When the pattern is simply a given string of symbols, there is an algorithm due to Morris, Knuth and Pratt which has a running time proportional to the total length of the pattern and long string together. This time may be achieved even on a Turing machine. The more difficult case where either string may have "don't care" symbols which are deemed to match with all symbols is also considered. By exploiting the formal similarity of string-matching with integer multiplication, a new algorithm has been obtained with a running time which is only slightly worse than linear. {AD 773-138}

TM-42
Rackoff, C.
ON THE COMPLEXITY OF THE THEORIES OF WEAK DIRECT PRODUCTS
Pages: 27 January 1974 $3.00
Keywords: Turing machine, mathematics, theory of computation
Abstract: Let N be the set of nonnegative integers and let $<N^*,+>$ be the weak direct product of $>N,+>$ with itself. Mostowski shows that the theory of $<N^*,+>$ is decidable, but his decision procedure isn't elementary recursive. We present here a more efficient procedure which operates within space $2^{2^{cn}}$. As corollaries we obtain the same upper bound for the theory of finite abelian groups, the theory of finitely generated abelian groups, and the theory of the structure $<N+,\cdot>$ of positive integers under multiplication. Fischer and Rabin have shown that the theory of $<N^*,+>$ requires time $2^{2^{2^{dn}}}$ on nondeterministic Turing machines.

We also obtain some very general results about the nature of the theory of the weak direct product of a structure with itself. {PB 228-459/AS}

TM-43
Fischer, M.J., Rabin, M.O.
SUPER-EXPONENTIAL COMPLEXITY OF PRESBURGER ARITHMETIC
Pages: 24 February 1974 $3.00
Keywords: Presburger arithmetic, mathematics, super-exponentiatial complexity
Abstract: Lower bounds are established on the computational complexity of the decision problem and on the inherent lengths of proofs for two classical decidable theories of logic: the first order theory of the real numbers under addition, and Presburger arithmetic — the first order theory of addition on the natural numbers. There is a fixed constant c > 0 such that for every (non-deterministic) decision procedure for determining the truth of sentences of real addition and for which the decision procedure runs for more than 2^{cn} steps. In the case of Presburger arithmetic, the corresponding bound is $2^{2^{cn}}$. These bounds apply also to the minimal lengths of proofs for any complete axiomatization in which the axioms are easily recognized. {AD 775-004}

TM-44
Pless, V.
SYMMETRY CODES AND THEIR INVARIANT SUBCODES
Pages: 13 February 1974 $3.00
Keywords: symmetry codes, invariant subcodes
Abstract:We define and study the invariant subcodes of the symmetry codes in order to be able to determine the algebraic properties of these codes. An infinite family of self-orthogonal rate 1/2 codes over GF(3), called symmetry codes, were constructed in [3]. A (2q + 2, q + 1) symmetry code, denoted by C(q), exists whenever q is an odd prime poiwer = -1, (mod 3). The group of monomial transformations leaving a symmetry code invariant is denoted by G(q). In this paper we construct two subcodes of C(q) denoted by R_σ(q) and R_μ(q). Every vector in R_σ(q) is invariant under a monomial transformation t in G(q) of odd order s where s divided (q + 1). Also R_μ(q) is invariant under t but not vector-wise. The dimensions of R_σ(q) and R_μ(q) are determined and relations between these subcodes are given. An isomorphism is constructed between R_σ and a subspace of W = V_3 2q + 2/s. It is shown that the image of R_σ is a self-orthogonal subspace of W. The isomorphic images of R_σ(17) (under an order 3 monomial) and R_S(29) (under an order 5 monomial) are both demonstrated to be equivalent to the (12, 6) Golay code. {AD 780-243}

TM-45
Fischer, M.J., Stockmeyer, L.J.
FAST ON-LINE INTEGER MULTIPLICATION
Pages: 23 May 1974 $3.00
Keywords: mathematics, integer multiplication, Turing machine
Abstract: A Turing machine multiplies binary integers *on-line* i it receives its inputs low-order digits first and produces the *j*th digit of the product before reading in the (*j* + 1)st digits of the two inputs. We present a general method for converting any off-line multiplication algorithm which forms the product of two *n*-digit binary numbers in time *F(n)* into an on-line method which uses time only 0(*F*(*n*) log *n*), assuming that *F* is monotone and satisfies $n \leq F(n) \leq F(2n)/2 \leq kF(n)$ for some constant *k*. Applying this technique to the fast multiplication algorithm of Schonhage and Strassen gives an upper bound of $0(n (\log n)^2 \log\log n)$ for on-line multiplication of integers. A refinement of the technique yields an optimal method for on-line multiplication by certain sparse integers. Other applications are to the on-line computation of products of polynomials, recognition of palindromes, and multiplication by a constant. {AD 779-889}

TM-46
Kedem, Z.M.
COMBINING DIMENSIONALITY AND RATE OF GROWTH ARGUMENTS FOR ESTABLISHING LOWER BOUNDS ON THE NUMBER OF MULTIPLICATIONS
Pages: 38 June 1974 $3.00
Keywords: algebrai manipulation, algorithms, computational complexity, lower bounds, polynomials, rate of growth, rational functions
Abstract: A new method for establishing lower bounds on the number of multiplications and divisions required to compute rational functions is described. The method is based on combining two known methods, dimensionality and rate of growth. The method is applied to several problems and new lower bounds. {PB 232-969/AS}

TM-47
Pless, V.
MATHEMATICAL FOUNDATIONS OF FLIP-FLOPS
Pages: 32 June 1974 $3.00
Keywords: mathematics, J-K flip-flops
Abstract: The main purpose of this paper is to lay a mathematical basis for the study of flip-flops. I feel that this approach will lead, in further studies, to important practical results although the alert reader can see some practical applications to this work. In this paper we only consider J-K flip-flops whose inputs are outputs of the other flip-flops {AD 780-901}

TM-48
Kedem, Z.M.
THE REDUCTION METHOD FOR ESTABLISHING LOWER BOUNDS ON THE NUMBER OF ADDITIONS
Pages: 19 June 1974 $3.00
Keywords: mathematics, algebraic manipulation, algorithms, computational complexity, rate of growth, rational functions, reduction
Abstract: A method for establishing lower bounds on the number of multiplications and divisions has been developed by Pan, Winograd and Strassen. A similar method is developed for establishing lower bounds on the number of additions and subtractions. The results obtained partially overlap those of Belaga, Winograd and Kirkpatrick. {PB 233-538/AS}

TM-49
Pless, V., Sloane, N.J.A.
COMPLETE CLASSIFICATION OF (24,12) AND (22,11) SELF-DUAL CODES
Pages: 39 June 1974 $3.00
Keywords: self dual codes
Abstract: A complete classification is given of all [22,11] and [24,12] self-dual codes. For each code we give the order of its group, the number of codes equivalent to it and its weight distribution. There is a unique [24, 12, 6] self-dual code. Several theorems on the enumeration of self-orthogonal codes are used, including formulas for the number of such codes with minimum distance ≥ 4, and for the sum of the weight enumerators of all self-dual codes. {AD 781-335}

TM-50
Benedict, G.G.
AN ENCIPHERING MODULE FOR MULTICS
Pages: 68 July 1974 $3.00
Keywords: Multics, enciphering, encryption
Abstract: Recently IBM Corporation has declassified an algorithm for encryption usable for computer-to-computer or computer-to-terminal communications. Their algorithm was implemented in a hardware device called Lucifer. A software implementation of Lucifer for Multics is described. A proof of the algorithm's reversibility for deciphering is provided. A special hand-coded (assembly language) version of Lucifer is described whose goal is to attain performance as close as possible to that of the hardware device. Performance measurements of this program are given. Questions addressed are: How complex is it to implement an algorithm is software designed primarily for digital hardware? Can such a program perform well enough for use in the I/O system of a; large time-sharing system? {AD 782-658}

TM-51
Aiello, J.M.
AN INVESTIGATION OF CURRENT LANGUAGE SUPPORT FOR THE DATA REQUIREMENTS OF STRUCTURED PROGRAMMING
Pages: 143 S.M. & E.E. Thesis/September 1974 $3.00
Keywords: structured programming, data structures, programming languages
Abstract: Structured programming is a new method for constructing reliable programs. Structured programming relies upon a systematic technique of top-down development which involves the refinement of both control structures and data structures. With possibly some limitations and extensions, existing languages can support control structures and data structures. With possibly some limitations and extensions, existing languages can support control structure refinement. On the other hand, it is the belief of many that the representation of data structure refinement cannot be satisfied by present-day languages. Before accepting this view, it is wise to explore its validity. Therefore this thesis will investigate whether existing languages with possibly slight modifications are adequate for supporting the data requirements of structured programming. {PB 236-815/AS}

TM-52
Lind, J.C.
COMPUTING IN LOGARITHMIC SPACE
Pages: 66 September 1974 $3.00
Keywords: logarithmic space, concatenation, recursion
Abstract: The set logspace, of logarithmic space computable string functions is defined. It is easily seen that logspace ⊃ polytime, the set of polynomial time computable functions. Logspace is shown to equal ***L***, the smallest class of recursive string functions containing concatenation and the equality function, and closed under explicit transformations, substitution of a function for a variable and two restricted types of recursion on notation. The first is called recursion of concatenation and only allows top level concatenation of the value of the recursive call. The second, called log bounded recursion on notation, will only define string functions whose length is bounded by 0(log n) on arguments of length n. Some additional closure properties of logspace are also described. {PB 236-815/AS}

TM-53
Bengelloun, S.A.
MDC-PROGRAMMER: A MUDDLE-TO-DATALANGUAGE TRANSLATOR FOR INFORMATION RETRIEVAL
Pages: 66 S.B. Thesis/October 1974 $3.00
Keywords: information retrieval, muddle
Abstract: This memo describes a practical application within the framework of the ARPA computer network of the philosophy that a fully developed computer network should appear as a virtual extension of the user's own software environment. The application involves the design and implementation of a software facility that will permit users at MIT's Dynamic Modeling System to consider the retrieval component of the Datacomputer (developed and run by the Computer Corporation of America) as an extension of the Muddle environment. This facility generates efficient Datalanguage retrieval code, handles inter-process control of the Datacomputrer, and manages all the necessary network connections. {AD 786-754}

TM-54
Meyer, A.R.
THE INHERENT COMPUTATION COMPLEXITY OF THEORIES OF ORDERED SETS: A BRIEF SURVEY
Pages: 10 October 1974 $3.00
Keywords: computational complexity, theories of ordered sets, theory of computation

Abstract: The significance of the theoretical distinctions between problems which are effectively decidable and those which are not can be challenged by objections of at least two kinds:

(1) Only a finite collection of sentences about arithmetic, for example, are of human concern, so the undecidability of the infinite collection of true sentences of arithmetic is immaterial.

(2) An efficient decision procedure for the mandic predicate calculus, for example, would have important practical applications, but the mere fact that it is effectively decidable is immaterial. {PB 237-200}

TM-55
Hsieh W.N., Harper, L.H., Savage, J.E.
A CLASS OF BOOLEAN FUNCTIONS WITH LINEAR COMBINATIONAL COMPLEXITY
Pages: 38 October 1974 $3.00
Keywords: combinational complexity, Boolean functions
Abstract: In this paper we investigate the combinational complexity of Boolean functions satisfying a certain property, $p^n_{k,m}$. A function of n variables has the $p^n_{k,m}$ property if there are at least m functions obtainable form each way of restricting it to a subset of n-k variables. We show that the complexity of a $p^n_{3,5}$ function is no less than 7n-4/6, and this bound cannot be much improved. Further, we find that for each k, there are $p^n_{k,2}$ k functions with complexity linear in n. {PB 237-206/AS}

TM-56
Gorry, G.A.
RESEARCH ON EXPERT SYSTEMS
Pages: 19 December 1974 $3.00
Keywords: expert systems
Abstract: To cope with the increasing complexity of social organizations and social processes, various segments of society have a growing need for experts. This paper discusses the need for and types of expert systems.

TM-57
Levin, M.
ON BATESON'S LOGICAL LEVELS OF LEARNING
Pages: 19 March 1975 $3.00
Keywords: learning, Batesons logic, social applications
Abstract: We live in a time when intellect seems to be responsible for the rapid intensification of horrible conditions that degrade the quality of life, and threaten to destroy the human species, and perhaps much of the biosphere of our planet along with it. What excited me about Bateson's approach was that it offers a view that is intellectually sound and scientific rather than romantic or anti-intellectual, and yet leads toward a balance or ecology of mind, rather than an intensification of the war between the familiar dualities: man vs nature, self vs. other, conscious vs. unconscious, life vs. death, subject vs. object. This essay is my attempt to work with this material in a personal way. {PB 237-033/AS}

TM-58
Qualitz, J.E.
DECIDABILITY OF EQUIVALENCE FOR A CLASS OF DATA FLOW SCHEMAS
Pages: 42 March 1975 $3.00
Keywords: data flow, decidability of equivalence, data flow schemas
Abstract: In this paper we examine a class of computation schemas and consider the problem of deciding when pairs of elements in this class represent equivalent programs. We are able to show that equivalence is decidable for a non-trivial class of unary operator data flow schemas, and consider the applicability of this result to the problem of deciding equivalence in related models of computation. {PB 237-033/AS}

TM-59
Hack, M.
DECISION PROBLEMS FOR PETRI NETS AND VECTOR ADDITION SYSTEMS
Pages: 79 March 1975 $3.00
Keywords: Petri nets, vector addition, decision problems
Abstract: Petri Nets, Generalized Petri Nets, and Vector Addition Systems can represent each other and thus have common decidability problems. The graphical appeal of Petri Nets is used in a new presentation of the classical problems of boundedness (decidable) and inclusion (undecidable). Various forms of the Reachability Problem are shown to be recursively equivalent to the Liveness Problem for Petri Nets. The decidability of these questions is still open, and some arguments both for and against the decidability of Liveness are presented. {PB 231-916/AS}

TM-60
Weiss, R.B.
CAMAC: GROUP MANIPULATION SYSTEM
Pages: 5 March 1975 $3.00
Keywords: group manipulation programs
Abstract: CMAC is a collection of group manipulation programs with an easy to use interface. With groups defined by either generating permutations or generators and relations the system can find coset tables, normalizers, centralizers, stabilizers, orbits, conjugacy classes, and isomorph classes of combinatorial objects, etc. {PB 240-495/AS}

TM-61
Dennis, J.B.
FIRST VERSION OF A DATA FLOW PROCEDURE LANGUAGE
Pages: 21 May 1975 $3.00
Keywords: data flow, functional languages, program graphs, data structures, heap, procedures, colored tokens
Abstract: A language for representing computational procedures based on the concept of data flow is presented in terms of a semantic model that permits concurrent execution of noninterfering program parts. Procedures in the language operate on elementary and structured values, and always define functional transformations of values. The language is equivalent in expressive power to a block structured language with internal procedure variables and is a generalization of pure Lisp. The language is being used as a model for study of fundamental semantic constructs for programming, as a target language for evaluating translatability of programs expressed at the user-language level and as a guide for research in advanced computer architecture.

TM-62
Patil, S.S.
AN ASYNCHRONOUS LOGIC ARRAY
Pages: 30 May 1975 $3.00
Keywords: logic design, asynchronous systems, logic arrays, digital systems, Petri nets, control structures
Abstract: A new asynchronous logic array for the general synthesis of asynchronous digital circuits is presented. The parallel and asynchronous nature of the array gives the realized systems the speed and characteristics of hardwired circuits even though they are implemented in a uniform diode array with appropriate terminating circuits. The logic array is particularly suited for implementing control structures and should help extend the field of micro-control to asynchronous and parallel computers.

TM-63
Pless, V.
ENCRYPTION SCHEMES FOR COMPUTER CONFIDENTIALITY
Pages: 19 May 1975 $3.00
Keywords: encryption, cryptography, enciphering, J-K flip-flops
Abstract: With the ever-increasing amount of data stored on computers, the need for security in transmission and storage becomes greater and greater. We here consider some new stream enciphering schemes based on J-K flip-flops. The data is considered to be a stream of binary bits. There are two main types of encipherment schemes; one is a block scheme which divides the data into blocks and then enciphers and deciphers a block at a time, the other is a stream scheme which enciphers and deciphers bit by bit. The stream enciphering scheme has the advantage that both the enciphering and the deciphering occur in real time. Since the aim of this paper is to present some new stream enciphering schemes, we shall describe briefly a general stream enciphering scheme. {AD A010-217}

TM-64
Weiss, R.B.
FINDING ISOMORPH CLASSES FOR COMBINATORIAL STRUCTURES
Pages: 23 S.M. Thesis/June 1975 $3.00
Keywords: combinatorial analysis, isomorph classes, rejection, complexity
Abstract: A common problem in combinatorial analysis is finding isomorph classes of combinatorial objects. This process is sometimes known as isomorph rejection. In graph theory, it is used to count labeled and unlabeled graphs with certain properties. In chemistry, it is used to count the number of structures with the same chemical formula. In computer science it is used in counting arguments in proofs in complexity theory. In coding theory, it is used to partition sets of vectors into easy to handle sets.

This thesis presents three different algorithms for solving this type of problem and compares their timing and memory use. Some examples are given of how to apply the algorithms to graph theory and coding theory.

TM-65
Fischer, M.J.
THE COMPLEXITY OF NEGATION-LIMITED NETWORKS- A BRIEF SURVEY
Pages: 12 June 1975 $3.00
Keywords: combinational complexity, mathematics, negation-limited networks
Abstract: Let B = (0,1), F_n = (f|f $B^n \mp B$), and let $\Omega \in \cup_{m \geq 1} F_m$. The combinational complexity $C^\Omega(F)$ of a set of Boolean functions F $\cap$ F_n is the least size network over the basis $MPF4W which computes each of the functions in F. Combinational complexity provides a meaningful measure of the difficulty of finite functions and has been widely studied. Our definitions are similar to those of Savage and are formalized in section 2.

Combinational complexity is interesting for both practical and theoretical reasons. The practical motivation comes form its correspondence with the cost of actual digital hardware. Theoretical interest derives both form its clean mathematical-structure and its connection with computation time on Turing machines. Namely, if g:B* $\mp$B* can be computed in time T(n) on a multitape Turing machine, then the restriction $g_n = g \mid B^n$ of g to length n inputs can be computed by a network over any complete basis of size 0(T(n) log T(n)).

It follows that a lower bound greater than cn log n on the combinational complexity of g_n implies a non-linear lower bound on the Turing machine time complexity of g. Such lower bounds on Turing machine time have never been obtained for particular concrete functions g except by diagonal techniques.

TM-66
Leung, C.K.C.
FORMAL PROPERTIES OF WELL-FORMED DATA FLOW SCHEMAS
Pages: 151 S.B., S.M. & E.E. Thesis/June 1975 $3.00
Keywords: data flow schemas, flow chart schemas, expressive power, decision problems
Abstract: This thesis presents some results in comparative schematology and some undecidability results for two models of computer programs: the class of flowchart schemas and the class of well-formed data flow schemas (wfdfs's). Algorithms are given for translating a schema in each class into an equivalent schema in the other class. The properties of freedom, α-freedom, openness and completeness are defined and studied. For every path P in a free flowchart schema S, there exists an interpretation under which the flow of control through S is along P. α-freedom is a generalization of freedom and captures the notion of freedom for wfdfs's. An open schema is one in which no basic component is redundant and a complete schema contains no subschema which, whenever enabled, does not terminate. A comparison of the expressive power of subclasses of flowchart schemas and wfdf's, possessing various combinations of these properties is made. It is shown that the class of free flowchart schemas properly contains the classes of free and α-free wfdfs's, and that the class of open and complete flowchart schemas is equivalent in expressive power to the class of open and complete wfdfs's. Three undecidability results for open and complete program schemas are established: openness is undecidable for open program schemas, and equivalence is undecidable for open program schemas, and equivalence is undecidable for open and complete program schemas.

TM-67
Cardoza, E.W.
COMPUTATIONAL COMPLEXITY OF THE WORLD PROBLEM FOR COMMUTATIVE SEMIGROUPS
Pages: 67 S.M. Thesis/October 1975 $3.00
Keywords: computational complexity, decision problems
Abstract: We analyze the computational complexity of some decision problems for commutative semigroups in terms of time and space on a Turing machine.

The main result we present is that any decision procedure for the word problem for commutative semigroups requires storage space at least proportional to n/logn on a multitape Turing machine. This implies that the word problem is polynomial space hard (and in particular that it is at least NP-hard).

We comment on the close relation of commutative semigroups to vector addition systems and Petri nets.

We also show that the lower bound of space n/logn can be extended to certain other natural algorithmic problems for commutative semigroups. Finally we show that for several other algorithmic problems for commutative semigroups there exist polynomial time algorithms.

TM-68
Weng, K.-S.
STREAM-ORIENTED COMPUTATION IN RECURSIVE DATA FLOW SCHEMAS
Pages: 93 S.M. Thesis/October 1975 $3.00
Keywords: parallel programming, applicative languages, data streams, data flow schemas, communicating module
Abstract: In this thesis we present a parallel programming language based on a parallel computation model known as data flow schemas. Syntactically, the language resembles programming languages such as Algol 60, but does not have GOTO's, WHILE-loops, and non-local variables. The attractiveness of this approach lies in the inherently determinate nature of data flow schemas and the possibility of formalizing the semantics of the language within the formalism suggested by Scott and Strachey. The language provides programming features for stream-oriented computation and intercommunicating systems. We introduce the notions of proper initialization and termination of such systems. A subclass of determinate systems in which these properties can be easily checked is defined and a translation into recursive data flow schemas is given.

TM-69
Bayer, P.J.
IMPROVED BOUNDS ON THE COSTS OF OPTIMAL AND BALANCED BINARY SEARCH TREES
Pages: 41 S.M. Thesis/November 1975 $3.00
Keywords: search trees, data storage

Abstract: A binary search tree can be used to store data in a computer system for retrieval by name. Different elements in the tree may be referenced with different probabilities. If we define the cost of the tree as the average number of elements which must be examined in searching for an element, then different trees have different costs. We show that two particular types of trees, weight balanced trees and min-max trees, which are easily constructed from the probability distribution on the elements are close to optimal. We gain added insight by deriving an expression for the expected value of the entropy of a random probability distribution.

TM-70
Ruth, G.R.
AUTOMATIC DESIGN OF DATA PROCESSING SYSTEMS
Pages: 25 FEBRUARY 1976 $3.00
Keywords: automatic programming, high level languages, Protosystem I
Abstract: The design of data organization and data accessing procedures for data processing systems operating on large keyed files of data is a common and recurrent activity in modern data processing applications. A considerable amount of understanding and expertise in this area has been developed and it is time to begin codifying and automating this process. It should be possible to develop a system where the user has merely to specify the characteristics of his data objects and their interrelations and the system will automatically determine the data organizations and accessing procedures that are optimal for his application. The optimizer for Protosystem I (an automatic programming system prototype at MIT) provides an example of how such automation can be accomplished. {AD A023-451}

TM-71
Rivest, R.
ON THE WORST-CASE OF BEHAVIOR OF STRING-SEARCHING ALGORITHMS
Pages: 8 April 1976 $3.00
Keywords: string-searching, pattern matching, computational complexity, worst-case performance
Abstract: Any algorithm for finding a pattern of length k in a string of length n must examine at least n-k + 1 of the characters of the string in the worst case. By considering the pattern 00...0, we prove that this is the best possible result. Therefore there do not exist pattern matching algorithms whose worst-case behavior is "sublinear" in n (that is, linear with constant less than one), in contrast with the situation for average behavior (the Boyer-Moore algorithm is known to be sublinear on the average.)

TM-72
Ruth, G.R.
PROTOSYSTEM I: AN AUTOMATIC PROGRAMMING SYSTEM PROTOTYPE
Pages: 27 July 1976 $3.00
Keywords: automatic programming, high level languages, Protosystem I
Abstract: A model of the data processing system writing process is given in terms of development stages. These stages correspond to the progression in the implementation and design process from the highest level of abstraction (English system specifications) to the lowest level (machine code). The issues and goals (including optimization of the product data processing systems) involved in automating these stages are discussed and strategies and methodologies used for doing so are developed.

Protosystem I, an automatic programming system prototype, is described. The completed (and working) part automates three of the five stages identified in the proposed model of the system writing process. The basic theory, methods and structure of this part of the automatic programming systems are presented. {AD A026-912}

TM-73
Rivest, R.
OPTIMAL ARRANGEMENT OF KEYS IN A HASH TABLE
Pages: 17 July 1976 $3.00
Keywords: hashing, collision resolution, searching, assignment problem, optimal algorithms, data base organization
Abstract: When open addressing is used to resolve collisions in a hash table, a given set of keys may be arranged in many ways; typically this depends on the order in which the keys are inserted. We show that arrangements minimizing either the average or worst-case number of probes required to retrieve any key in the table can be found using an algorithm for the assignment problem. The worst-case retrieval time can be reduced to $0(\log_2(M))$ with probability $1-\epsilon(M)$, when storing M keys in a table of size M, where $\epsilon(M) \mp 0$ as $M \mp \infty$. We also examine insertion algorithms to see how to apply these ideas for a dynamically changing set of keys.

TM-74
Malvania, N.
THE DESIGN OF A MODULAR LABORATORY FOR CONTROL ROBOTICS
Pages: 162 S.M. Thesis/September 1976 $3.00
Keywords: process control, computer control, digital systems, real-time systems, control laboratory
Abstract: Computers have been used for the control of physical processes since the early sixties. In this thesis, we look at Control Robotics, the procedural control of physical processes. Based upon this new approach, a design for a modular laboratory is proposed. The laboratory consists of a set of experiments which can be synthesized using certain conversion and processing modules. The laboratory also entails the generation of algorithms and programs for each experiment. Experiments are proposed and analyzed, and a common and in a sense minimal set of hardware modules is selected using a minimax approach. Power, torque, strength, resolution and other similar requirements for the modules are discussed. A theoretical model is developed for predicting and analyzing the capability of a processor to perform real-time control. The model is based upon the so-called Earliset Deadline algorithm for scheduling a number of tasks on a single processor. The model relates the bandwidths of different tasks a processor can perform to the total number of tasks; the average instruction execution time for the processor; and the complexity of its instruction set. This model is used to exhibit and compare the controlling capacities of two processors - Digital Equipment Corporation's PDP 11/45 and Intel 8080. It is also used to predict the processor requirements for the experiments of the proposed modular laboratory. Thesis results include measures of relative power of the tested processors in the context of real-time control, and their capabilities in carrying out the experiments of the proposed laboratory. {AD A030-418}

TM-75
Yao, A.C., Rivest, R.
K + 1 HEADS ARE BETTER THAN K
Pages: 8 September 1976 $3.00
Keywords: multihead finite automata
Abstract: There are languages which can be recognized by a deterministic (k + 1)-headed one-way finite automaton but which cannot be recognized by a k-headed one-way (deterministic or non-deterministic) finite automaton. Furthermore, there is a language accepted by a 2-headed nondeterministic finite automaton which is accepted by no k-headed deterministic finite automaton. {AD A030-008}

TM-76
Bloniarz, P.A., Fischer, M.J., Meyer, A.R.
A NOTE ON THE AVERAGE TIME TO COMPUTE TRANSITIVE CLOSURES
Pages: 10 September 1976 $3.00
Keywords: transitive closure, Spiras algorithm
Abstract: An algorithm which finds shortest paths between all pairs of nodes in an n node weighted, directed graph using an average of $0(n^2*(\log n)^2)$ basic steps has been described by Spira. A special case of the shortest path problem is the transitive closure problem for Boolean matrices.

In this note we point out a simple restriction of Spira's algorithm which allows the computation of the transitive closure of a Boolean matrix in average time $0(n^2*\log m)$. (This time bound for the average case was obtained independently by D. Angluin using a different algorithm.) In the course of verifying the restricted algorithm, we isolate a lacuna in Spira's original procedure - namely Spira's algorithm does not specify from which node to search when several nodes are equidistant from a source. We describe a counter-example based on this lacuna showing that Spira's algorithm may run in $\Omega(n^3)$ average steps on certain ensembles of graphs when "tie-breaking" in the case of equidistant nodes is decided by a plausible but improper convention. With a proper tie-breaking procedure, Spira's algorithm indeed can be shown to run in $0(n^2*(\log n)^2)$ steps on the average for a somewhat larger class of probability measures on graphs than he originally claimed, although we do not prove this latter fact in the present note.

TM-77
Mok, A.K.
TASK SCHEDULING IN THE CONTROL ROBOTICS ENVIRONMENT
Pages: 90 S.M. Thesis/September 1976 $3.00
Keywords: real-time systems, scheduling, process control, multiprocessor scheduling, robotics
Abstract: Scheduling problems involved in Control Robotics, a software approach to control engineering are studied. The capability of a multiprocessor system to handle tasks with hard, real-time deadlines is in-

vestigated according to whether complete or partial a priori knowledge of the deadlines, computation times and frequencies of occurrence of individual tasks is available. A model of preemptive scheduling, the "scheduling game" is introduced to explore mathematical relationships for different scheduling situations. A necessary and sufficient condition for scheduling tasks with simultaneous requests or deadlines is derived. Partial solutions and the difficulties involved in scheduling tasks with distributed requests are discussed. It is shown that in the most general case, there is no globally optimal algorithm in the absence of a priori knowledge about the distribution of requests of future tasks in time. {AD A030-402}

TM-78
Benjamin, A.J.
IMPROVING INFORMATION STORAGE RELIABILITY USING A DATA NETWORK
Pages: 136 S.M. Thesis/October 1976 $3.00
Keywords: reliability, backup, distributed systems, remote storage
Abstract: Backup and recovery methods using magnetic tapes are common in computer utilities, since information stored on-line is subject to damage. The serial access nature of the tape medium severely restricts the flexibility and simplicity of accessing and managing the stored data. A method using a data network will be described, to present a backup mechanism which takes advantage of a large, inexpensive, random access remote data storage facility to provide data access and management functions that are more flexible than those provided by a traditional backup facility. Although data transfer rates will be reduced, data access and management will be simplified, and system availability will be improved. The work described is based on a network backup facility built for the Multics computer utility, using the ARPAnet. {AD A033-394}

TM-79
Brown, G.P.
A SYSTEM TO PROCESS DIALOGUE: A PROGRESS REPORT
Pages. 52 October 1976 $3.00
Keywords: dialogue, English understanding, natural language semantic network
Abstract: This is a progress report on work toward an English language interface for expert systems. A framework for handling mixed-initiative English dialogue in a console session environment is disucssed, with special emphasis placed on recognition. The ideas presented here are being implemented in a prototype system called Susie Software, which is embedded in the OWL system. OWL is currently under development in the Automatic Programming Group at the MIT Laboratory for Computer Science. We are using OWL to explore the problems of constructing expert systems, and for Susie Software the domain of expertise is programming. In the Susie effort to date, major emphasis has been placed on the construction of a computational model for the structural aspects of English dialogue; it is this structural model that will be discussed. {AD A033-276}

TM-80
Even, S.
THE MAX FLOW ALGORITHM OF DINIC AND KARZANOV: AN EXPOSITION
Pages: 13 December 1976 $3.00
Keywords: Karzanov algorithm, network flow
Abstract: Recently A.V. Karzanov improved Dinic's algorithm to run in time $0(n^3)$ for networks of n vertices. For the benefit of those who do not read Russian, the Dinic-Karzanov algorithm is explained and proved.

In addition to being the best algorithm known for network flow, this algorithm is unique in that it does not use path augmentation.

TM-81
Gifford, D.
HARDWARE ESTIMATION OF A PROCESS' PRIMARY MEMORY REQUIREMENTS
Pages: 56 S.M. Thesis/January 1977 $3.00
Keywords: primary memory requirement, virtual memory, level of multiprogramming, working set, associative memory, resource allocation, LRU stack model, referencing characteristics
Abstract: It is shown that a process' primary memory requirements can be approximated by use of the miss rate in the Honeywell 6180's page table word associative memory. This primary memory requirement estimate was employed by an experimental version of Multics to control the level of multiprogramming in the system, and bill for memory usage. The resultant system's tuning parameters were shown to be configuration insensitive, and it was conjectured that the system would also track shifts in the referencing characteristics of its workload and keep the system in tune. The limitations of the assumptions made about a process' referencing characteristics are examined, and directions for future research are outlined.

TM-82
Rivest, R., Shamir, A., Adelman, L.M.
A METHOD FOR OBTAINING SIGNATURES AND PUBLIC-KEY CRYPTOSYSTEMS
Pages: 15 April 1977 $3.00
Keywords: digital signatures, public-key cryptosystems, privacy, authentication, security, factorization, prime numbers, electronic mail, message passing, electronic funds transfer, cryptography
Abstract: We present an encryption method with the novel property that publicly revealing an *encryption* key does not thereby reveal the corresponding *decryption* key. This has two important consequences:

(1) Couriers or other *secure* means are not needed to transmit keys, since a message can be enciphered using an encryption key publicly revealed by the intended recipient. Only he can decipher the message, since only he knows the corresponding decryption key.

(2) A message can be "signed" using a privately-held decryption key. Anyone can verify this signature using the corresponding publicly revealed encryption key. Signatures cannot be forged, and a signer cannot later deny the validity of his signature. This has obvious applications in "electronic mail" and "electronic funds transfer" systems.

A message is encrypted by representing it as a number M, raising M to a publicly-specified power *e*, and then taking the remainder when the result is divided by the publicly specified product *n*n of two large secret prime numbers *p* and *q*. Decryption is similar; only a different, secret, power *d* is used, where $e*d \equiv 1(\bmod(p-1)*(q-1))$. The security of the system rests in part on the difficulty of factoring the published divisor, *n*. {AD A039-036}

TM-83
Baratz, A.E.
CONSTRUCTION AND ANALYSIS OF NETWORK FLOW PROBLEM WHICH FORCES KARZANOV ALGORITHM TO $0(n^3)$ RUNNING TIME
Pages: 17 April 1977 $3.00
Keywords: network flow, Karzanov algorithm
Abstract: The intent of this paper is to demonstrate the construction of a network flow problem which will force the Karzanov "Preflow" algorithm to run its theoretic worst case time $0(n^3)$. Once such a "bad case" network has been constructed, an analysis is performed to determine the exact time required by the algorithm to compute the maximum flow through the network.

TM-84
Rivest, R., Pratt, V.R.
THE MUTUAL EXCLUSION PROBLEM FOR UNRELIABLE PROCESSES
Pages: 28 April 1977 $3.00
Keywords: mutual exclusion problem, unreliable processes
Abstract: Consider n processes operating asynchronously in parallel, each of which maintains a single "public" variable which can be read (but not written) by the other process. We show that the process can synchronize their actions by the basic operations of (1) reading each other's public variables, and (2) setting their own public variable to some value. A process may "die" (fail) at any time, when its public variable is (automatically) set to a special "dead" value. A dead process may revive. Reading a public variable which is being simultaneously updated returns either the old or the new value.

Each process may be in a certain "critical" state (which it leaves if it dies). We present a synchronization scheme with the following properties.

(1) At most one process is ever in its critical state at a time.

(2) If a process wants to enter its critical state, it may do so before any other process enters its critical state more than once.

(3) The public variables assume only a finite number of values.

(4) A process wanting to enter its critical state can always make progress towards that goal.

(5) The various processes may run at arbitrary speeds relative to one another.

By the definition of the problem, no process can prevent another form entering its critical state by repeatedly failing and restarting.

In the case of two processes, what makes our solution of particular interest is its remarkable simplicity when compared with the extant solutions to this problem. Our n-process solution uses the two-process solution as a subroutine, and is not quite as elegant as the two-process solution.

TM-85
Shamir, A.
FINDING MINIMUM CUTSETS IN REDUCIBLE GRAPHS
Pages: 25 June 1977 $3.00
Keywords: minimum cutsets, program verification, reducible graphs
Abstract: The analysis of many processes modeled by directed graphs requires the selection of a subset of vertices which cut all the cycles in the graph. Reducing the size of such a cutset usually leads to a simpler and more efficient analysis, but the problem of finding minimum cutsets in general directed graphs is known to be NP-complete. In this paper we show that in reducible graphs (and thus in almost all the "practical" flowcharts of programs), minimum cutsets can be found in linear time. An immediate application of this result is in program verification systems based on Floyd's inductive assertions method. {AD A040-698}

TM-86
Szolovits, P., Hawkinson, L.B., Martin, W.A.
AN OVERVIEW OF OWL, A LANGUAGE FOR KNOWLEDGE REPRESENTATION
Pages: 24 June 1977 $3.00
Keywords: artificial intelligence, knowledge representation, LMS, memorystructures, natural language, OWL, symbolic manipulation
Abstract: We describe the motivation and overall organization of the OWL language for knowledge representation. OWL consists of a memory of concepts in terms of which all English phrases and all knowledge of an application domain are represented, a theory of English grammar which tells how to map English phrases into concepts, a parser to perform that mapping for individual sentences, and an interpreter to carry out procedures which are written in the same representational formalism. The system has been applied to the study of interactive dialogs, explanations of its own reasoning, and question answering. {AD A041-372}

TM-87
Clark, D.D., Editor
ANCILLARY REPORTS: KERNEL DESIGN PROJECT
Pages: 105 June 1977 $3.00
Keywords: Multics, security kernel, operating systems
Abstract: For the past three years, the Computer Systems Research Division of the Laboratory for Computer Science has performed a series of engineering studies on the Multics operating system. The goal was to demonstrate the feasibility of producing a version of a full function general purpose operating system with a "security kernel" simple enough that its correct operating can be certified by some form of auditing. During this project, several results of an interim nature were published as internal group memos, and were never subsequently published in any publicly available form. This memo contains seven such reports that contain interesting results not otherwise reported. These seven reports deal with four areas:

* Analysis of bugs discovered in the Multics system.
* Survey of the initial size of the Multics kernel.
* Detailed design specification of two level process manager.
* Performance evaluation of the multi-process page manager.

TM-88
Lloyd, E.L.
ON TRIANGULATIONS OF A SET OF POINTS IN THE PLANE
Pages: 56 S.M. Thesis/July 1977 $3.00
Keywords: mathematics, NP-complete, triangulation
Abstract: A set, V, of points in the plane is triangulated by a subset, T, of the traight line segments whose endpoints are in V, if T is a maximal subset such that the line segments in T intersect only at their endpoints. The weight of any triangulation is the sum of the Euclidean lengths of the line segments in the triangulation. We examine two problems involving triangulations. We discuss several aspects of the problem of finding a minimum weight triangulation among all triangulations of a set of points and give counterexamples to two published solutions to this problem. Secondly, we show that the problem of determining the existence of a triangulation in a given subset of the straight line segments whose endpoints are in V is NP-Complete.

TM-89
Rodriguez, H.
MEASURING USER CHARACTERISTICS ON THE MULTICS SYSTEM
Pages: 52 S.B. Thesis/August 1977 $3.00
Keywords: Multics, performance
Abstract: One of the problems in measuring the performance of a computer system is in defining its normal workload. In the case of timesharing systems, it is necessary to develop a behavioral model of the average user. This thesis presents a study of several parameters that characterize user behavior on monitoring the logon sessions of three different groups of users. The results are presented and comparisons are made between the command usage of the groups. Some patterns of usage do appear in the results, but it is unclear if they can be applied in other situations.

A probability distribution of the think time between commands is shown and compared with other distributions. The benchmark program currently used on the Multics system is also compared with the user model described in this study. The capability to monitor user behavior and characteristics is shown to be useful and worth installing in the system.

TM-90
dOliveira, C.R.
AN ANALYSIS OF COMPUTER DECENTRALIZATION
Pages: 60 S.B. Thesis/October 1977 $3.00
Keywords: computer decentralization
Abstract: This thesis is concerned with the recent trend towards decentralization of the computer facility. We conjecture that there are strong forces in many organizations leading towards decentralization, which have been held in check by technological and economic constraints that are beginning to relax. This conjecture is explored by analyzing approximately forty case studies of decentralization decisions.

The results indicate that (1) strong decentralization forces do exist in many organizations. The forces derived from these particular case studies are classified as either functional, economic or psychological. (2) The drop in hardware costs allows decentralization to occur at the initiative of lower level managers.

The consequences could include disintegration of the organization's information system. Decisions by lower level managers may overlook the technological constraints of decentralization, especially the problems of networking loosely coupled computers. This could result in a future inability to share data or programs among organizational units. Because of the many functional advantages it provides, we do not feel that top level management should discourage decentralization. However, top level management must be aware that the technological constraints require that decentralization occur with their guidance and their perspective of the entire organization. {AD A045-526}

TM-91
Shamir, A.
FACTORING NUMBERS IN 0 (LOG n) ARITHMETIC STEPS
Pages: 13 November 1977 $3.00
Keywords: arithmetical, mathematics, factorization, prime numbers
Abstract: In this paper we show that a non-trivial factor of a composite number n can be found by performing arithmetic steps in a number proportional to the number of bits in n, and thus there are extremely short straight-line factoring programs. However, this theoretical result does not imply that natural numbers can be factored in polynomial time in the Turing-Machine model of complexity, since the numbers operated on can be as big as 2^{cn^2}, thus requiring exponentially many bit operations. {AD A047-709}

TM-92
Misunas, D.P.
REPORT ON THE WORKSHOP ON DATA FLOW COMPUTER AND PROGRAM ORGANIZATION
Pages: 50 November 1977 $3.00
Keywords: data flow, computer architecture, workshop report, performance, simulation, specification, verification, architecture
Abstract: This report comprises an edited transcription of presentations made at the Workshop on Data Flow Computer and Program Organization, held at M.I.T. on July 10-14, 1977 and co-sponsored by the Lawrence Livermore Laboratory (LLL) and the Department of Energy, Mathematical Sciences Branch. These informal transcriptions are only intended to provide a general picture of ongoing work in the area and, to that end, have been heavily edited and summarized. For further details, the interested reader should consult the bibliography at the end of the report.

TM-93
Amikura, K.
A LOGIC DESIGN FOR THE CELL BLOCK OF A DATA FLOW PROCESSOR
Pages: 103 S.M. Thesis/December 1977 $3.00
Keywords: data flow, processor design, computer architecture, architecture, modular design, asynchronous systems, Petri nets

Abstract: Recently studies on parallel computation architecture have yielded a new type of computer architecture known as the data flow processor. As part of the effort in realizing the data flow processor, a logic design for the Cell Block of the basic data flow processor is proposed in this thesis. The resulting design has a modular structure which is derived from a top-down decomposition of the specification given in an Architecture Description Language. The desired speed of operation of the Cell Block is obtained by exploiting the parallelism inherent in its operation. The logic design is carried out using electronic devices available commercially today, but is based on an asynchronous communication protocol.

TM-94
Berez, J.M.
A DYNAMIC DEBUGGING SYSTEM FOR MDL
Pages: 53 S.B Thesis/January 1978 $3.00
Keywords: programming languages, debugging, graphics, MDL
Abstract: Program debugging is a time consuming process. Conventional debugging techniques and aids typically give the user a narrow view of the program's operation, making debugging difficult. A debugging system that would present a clear overall picture of a program's behavior and would be both flexible and simple to operate would be a valuable tool. Such a system was designed and implemented in and for MDL, a high-level applicative programming language. This report discusses: the design alternatives considered during the debugging system's design and implementation phases, the reasons for the resulting design choices, and the system attributes. A major attribute of the system (MEND) is that it does not simulate the program being debugged but instead monitors it from another process. This attribute results in a robust and viable debugging system, because MEND need not be modified in order to handle each new extension to MDL and/or each new user-defined primitive. {AD A050-191}

TM-95
Harel, D.
CHARACTERIZING SECOND ORDER LOGIC WITH FIRST ORDER QUANTIFIERS
Pages: 7 March 1977 $3.00
Keywords: first order logic, Henkin prefix, partially ordered quantifiers, second order logic
Abstract: A language Q is defined and given semantics, the formulae of which are quantifier-free first-order matrices prefixed by combinations of finite partially ordered first-order quantifiers. It is shown that Q is equivalent in expressive power to second order logic by establishing the equivalence of alternating second order quantifiers and forming conjunctions of partially ordered first-order quantifiers.

TM-96
Harel, D., Pnueli, A., Stavi, J.
A COMPLETE AXIOMATIC SYSTEM FOR PROVING DEDUCTIONS ABOUT RECURSIVE PROGRAMS
Pages: 24 February 1978 $3.00
Keywords: Hoare logic, partial correctness, recursive programs, relative completeness, total correctness
Abstract: Denoting a version of Hoare's system for providing partial correctness of recursive programs by ***H***, we present an extension ***D*** which may be thought of as $H \cup \{<,>,\mathrm{E},\mathrm{A}\} > \cup H^{-1}$, including the rules of ***H***, four special purpose rules and inverse rules to those of Hoare. ***D*** is shown to be a complete system (in Cook's sense) for proving deductions of the form $\sigma_1,...,\sigma_n$ σ over a language, the wff's of which are assertions in some assertion language L and partial correctness specifications of the form $p\{\alpha\}q$. All valid formulae of L are taken as axioms of ***D***. It is shown that ***D*** is sufficient for proving partial correctness, total correctness and program equivalence as well as other important properties of programs, the proofs of which are impossible in ***H***. The entire presentation is worked out in the framework of nondeterministic programs employing iteration and mutually recursive procedures.

TM-97
Harel, D., Meyer, A.R., Pratt, V.R.
COMPUTABILITY AND COMPLETENESS INLOGICS OF PROGRAMS .br
Pages: 16 February 1978 $3.00
Keywords: arithmetical, dynamic logic, r.e. programs, regular programs, relative completeness, validity problem
Abstract: Dynamic logic is a generalization of first order logic in which quantifiers of the form "for all X..." are replaced by phrases of the form "after executing program α...". This logic subsumes most existing first-order logics of programs that manipulate their environment, including Floyd's and Hoare's logics of partial correctness and Manna and Waldinger's logic of total correctness, yet is more closely related to classical first-order logic than any other proposed logic of programs. We consider two issues: how hard is the validity problem for the formulae of dynamic logic, and how might one axiomatize dynamic logic? We give bounds on the validity problem for some special cases, including a Π_2^0-completeness result for the partial correctness theories of uninterpreted flowchart programs and a Π_1^1-completeness result for unrestricted dynamic logic. We also demonstrate the completeness of an axiomatization of dynamic logic relative to arithmetic.

TM-98
Harel, D., Pratt, V.R.
NONDETERMINISM IN LOGICS OF PROGRAMS
Pages: 27 February 1978 $3 00
Keywords: arithmetical, divergence, dynamic logic, failure, nondeterminism, total correctness, weakest preconditions
Abstract: We investigate the principles underlying reasoning about nondeterministic programs, and present a logic to support this kind of reasoning. Our logic, an extension of dynamic and subsumes most existing first-order logics of nondeterministic programs, including that development by Dijkstra based on the concept of weakest precondition. A significant feature is the strict separation between the two kinds of nonterminating computations: infinite computations and failures The logic has a Tarskian truth-value semantics, an essential prerequisite to establishing completeness of axiomatizations of the logic. We give an axiomatization for flowchart (regular) programs that is complete relative to arithmetic in the sense of Cook. Having a satisfactory tool at hand, we turn to the clarification of the concept of the total correctness of nondeterministic programs, providing in passing, a critical evaluation of the widely used "predicate transformer" approach to the definition of programming constructs, initiated by Dijkstra. Our axiom system supplies a complete axiomatization of *wp*.

TM-99
LaPaugh, A.S.
THE SUBGRAPH HOMEOMORPHISM PROBLEM
Pages: 118 S.M. Thesis/February 1978 $3.00
Keywords: homeomorphisms of graphs, pathfinding algorithms, forbidden subgraph properties, cycles
Abstract: The problem investigated in this thesis is that of finding homeomorphic images of a given graph, called the pattern graph, in a larger graph. A homeomorphism is a pair of mappings, (ν, α), such that ν maps the nodes of the pattern graph to nodes of the larger graph, and α maps the edges of the pattern graph to (edge or node) disjoint paths in the larger graph. A homeomorphism represents a similarity of structure between the graphs involved. Therefore, it is an important concept for both graph theory and applications such as programming schema.

We give a formal definition of the subgraph homeomorphism problem. In our investigation, we focus on algorithms which depend on the pattern graph and allow the node mapping, ν, to be partially or totally specified. Reductions between node disjoint and edge disjoint formulations of the problem are discussed. Also, reductions facilitating the solution of given subgraph homeomorphism problems are formulated. A linear time algorithm for finding a cycle in a graph containing three given nodes of the graph is presented. Finally, the two disjoint paths problem, an open problem, is discussed in detail.

TM-100
Misunas, D.P.
A COMPUTER ARCHITECTURE FOR DATA FLOW COMPUTATION
Pages: 107 S.M. Thesis/March 1978 $3.00
Keywords: computer architecture, data flow, parallelism
Abstract: The structure of a computer which utilizes a data flow program representation as its base language is described. The use of the data flow representation allows full exploitation by the processor of the parallelism and concurrency achievable through the data flow form. The unique architecture of the processor avoids the usual problems of processor switching and memory/processor interconnection by the use of interconnection networks which have a great deal of inherent parallelism. The structure of the processor allows a large number of instructions to be active simultaneously. These active instructions pass through the interconnection networks concurrently and form streams of instructions for the pipelined functional units.

Due to the cyclic nature of an iterative computation, the possibility of deadlock can arise in the performance of such a computation within data flow architecture. A deadlock is caused by the interaction of several simultaneously active cycles of the same iterative computation. The use of a recursive rather than iterative representation of a computation avoids the deadlock problem and provides a more efficient implementation of the computation within the architecture. For this reason, a program executed by the data flow processor is restricted to an acyclic directed graph representation. {AD A052-538}

TM-101
Martin, W.A.
DESCRIPTIONS AND THE SPECIALIZATION OF CONCEPTS
Pages: 37 March 1978 $3.00
Keywords: artificial intelligence, knowledge representation, semantics, OWL
Abstract: The OWL II system computes with expressions which describe an object from a particular viewpoint. These partial descriptions form a tree structure under the specialization operation, which preserves intentional properties. The descriptions are also related in terms of their extensions by characterization and exemplar links. Descriptions of individuals must always specify a context of the description. Eight ways in which one description can be a specialization of another are distinguished. {AD A052-773}

TM-102
Abelson, H.
LOWER BOUNDS ON INFORMATION TRANSFER IN DISTRIBUTED COMPUTATIONS
Pages: 14 April 1978 $3.00
Keywords: computational complexity, distributed processing
Abstract: We derive a lower bound on the interprocessor information transfer required for computing a function in a distributed network. The bound is expressed in terms of the function's derivatives, and we use it to exhibit functions whose computation requires a great deal of interprocess communication. As a sample application, we give lower bounds on information transfer in the distributed computation of some typical matrix operations.

TM-103
Harel, D.
ARITHMETICAL COMPLETENESS IN LOGICS OF PROGRAMS
Pages: 26 APRIL 1978 $3.00
Keywords: arithmetical, divergence, dynamic logic, relative completeness
Abstract: We consider the problem of designing *arithmetically complete* axiom systems for proving general properties of programs; i.e. axiom systems which are complete over *arithmetical universes*, when all first-order formulae which are valid in such universes are taken as axioms. We prove a general Theorem of Completeness which takes care of a major part of the responsibility when designing such systems. It is then shown that what is left to do in order to establish an arithmetical completeness result, such as those appearing in [12] and [14] for the logics DL and DL^+, can be described as a chain of reasoning which involves some simple utilizations of arithmetical induction. An immediate application of these observations is given in the form of an arithmetical induction. An immediate application of these observations is given in the form of an arithmetical completeness result for a new logic similar to that of Salwicki. Finally, we contrast this discipline with Cook's notion of *relative* completeness.

TM-104
Jaffe, J.M.
THE USE OF QUEUES IN THE PARALLEL DATA FLOW EVALUATION OF "IF-THEN-WHILE" PROGRAMS
Pages: 22 May 1978 $3.00
Keywords: computation graphs, data dependency programs, data flow computation, if-then-while programs, parallel computation, parallel schemata, queues,data flow schema
Abstract: A property of a model of parallel computation is analyzed. We show that the use of queues may speed-up the execution of well formed data flow schemas by an arbitrarily large factor. A general model of data flow computation is presented to provide a framework for the comparison of data flow models. In particular a formal definition of a data flow version of the Computation Graphs of Karp and Miller and the Data Flow Schemas of Dennis are provided within the context of this model.

TM-105
Masek, W.J., Paterson, M.S.
A FASTER ALGORITHM COMPUTING STRING EDIT DISTANCES
Pages: 26 May 1978 $3.00
Keywords: string editing, longest common subsequence
Abstract: The edit-distance between two character strings can be defined as the minimum cost of a sequence of editing operations which transforms one string into the other. The operations allowed are deleting, inserting and replacing one symbol at a time, with possibly different costs for each of these operations. The problem of finding the longest common subsequence of two strings is a special case of the problem of computing edit-distances.

We describe an algorithm for computing the edit-distance between two strings of length n and m, $n \geq m$, which requires 0(nm/min(log n, m)) steps whenever the costs of edit-operations are integral multiples of a single positive real number and the alphabet for the strings is finite. These conditions are necessary for the algorithm to achieve the time bound.

TM-106
Parikh, R.
A COMPLETENESS RESULT FOR A PROPOSITIONAL DYNAMIC LOGIC
Pages: 19 July 1978 $3.00
Keywords: completeness, dynamic logic, modal logic, program semantics
Abstract: Propositional modal logic of programs has been introduced by Fischer and Ladner, following the ideas of Pratt. We shall call it propositional dynamic logic (PDL) following the terminology of Harel, Meyer and Pratt. In the following we prove the completeness of a rather natural set of axioms for this logic and for an extension of it obtained by allowing the inverse operation which converts a program into its inverse.

The following is a brief sketch of the plan of the proof. We introduce two auxiliary notions, that of pseudomodel and that of nonstandard model. Pseudomodels are highly syntactic objects and merely represent partial attempts to spell out a model. Thus an inconsistent formula may have a pseudomodel but every attempt to spell out the complete details of a model corresponding to the pseudomodel will, for an inconsistent formula, run into obstacles. A nonstandard model is like a model but we do not insist that R_{α^*} ^{5} $(R_\alpha)^*$. R_{α^*} is some reflexive transitive relation containing R_α, but not necessarily the smallest.

We shall show that if a formula A is not disprovable from the axioms then it has a series of consistent pseudomodels whose union is a nonstandard model satisfying certain special induction axioms. It is then shown how such a nonstandard model can be converted into a model in the usual sense.

TM-107
Shamir, A.
A FAST SIGNATURE SCHEME
Pages: 17 July 1978 $3.00
Keywords: electronic signatures, public-key cryptosystems, knapsack problems, coding techniques
Abstract: In this paper we propose a new scheme for generating and verifying "electronic signatures" in public-key communications. The scheme is based on the difficulty of solving the knapsack problem, and its tow main advantages over previous schemes are speed and simplicity. {ADA057-152}

TM-108
Baratz, A.E.
AN ANALYSIS OF THE SOLOVAY AND STRASSEN TEST FOR PRIMALITY
Pages: 22 July 1978 $3.00
Keywords: Carmichael number, Jacobi symbol, primality, probabilistic algorithms, quadratic residue
Abstract: In this paper we will analyze the performance of the Solvay and Strassen probabilistic primality testing algorithm. We will show that iterating Solovay and Strassen's algorithm r times, using independent random numbers at each iteration, results in a test for the primality of any positive odd integer, $n > 2$, with error probability 0 (if n is prime), error probability at most 4^{-r}(if n is composite and non-Carmichael), and error probability at most 2^{-r} (if n is composite and Carmichael).

TM-109
Parikh, R.
EFFECTIVENESS
Pages: 17 July 1978 $3.00
Keywords: effectiveness, Turing machine, program correctness, grammar

Abstract: Church's thesis equates the intuitive notion *effective* with the mathematical notion *recursive*. In order for this thesis to provide any information to us we have to have clear understanding of both notions. We consider one of the prevalent definitions of *effective* and compare it with the notions of syntactic and semantic consequence to see which one it corresponds to better. The notion of syntactic consequence, while useful, is subservient to the semantic notion and when we go from one language to another we expect to have to change the syntactic notion of consequence, if we are lucky enough to have one at all. Similarly the prevalent notion of effectiveness is a restricted one and has had the effect of limiting our view. At the end of section 3, we give a more general analysis of effectiveness and propose a mathematical theory. In section 4 we consider the question whether the set of grammatical sentences of English is recursive. We show that this question is not well posed and that the arguments in favor of a positive answer are question begging. We reformulate this question in the form "How recursive is the set of grammatical sentences of English?", and propose a way of turning it into a precise technical problem. The method used is a generalization of the Kolmogorov-Chaitin theory of randomness which is briefly sketched.

TM-110
Jaffe, J.M.
AN ANALYSIS OF PREEMPTIVE MULTIPROCESSOR JOB SCHEDULING
Pages: 11 September 1978 $3.00
Keywords: scheduling, maximal usage schedules, worst case performance, preemption
Abstract: The preemptive scheduling of a partially ordered set of tasks is studied. A class of scheduling heuristics is introduced, and the performance of schedules in this class is analyzed with respect to the least finishing time optimality criterion. If there are *m* processors, then the finishing time of any schedule in the class is at most (sq.rt. *m*) + (1/2) times worse than optimal, independent of the speeds of the processors. Examples are given which indicate that there are schedules which may be as bad as (sq.rt. *m*-1) times worse than optimal even for machines with one fast processor.

TM-111
Jaffe, J.M.
BOUNDS ON THE SCHEDULING OF TYPED TASK SYSTEMS
Pages: 25 September 1978 $3.00
Keywords: scheduling, list scheduling, typed task systems, data flow computation, worst case performance
Abstract: We study the scheduling of different types of tasks on different types of processors. If there are *k* types of tasks and m_i identical processors for tasks of type *i*, the finishing time of any demand driven or list schedule is at most $k+1$-(1/max(m_1,...,m_k)) times worse than the optimal schedule. This bound is best possible. If the processors execute at different speeds then the performance ratio of any list schedule (relative to the optimal schedule) is bound by *k* plus the maximum ratio between the speeds of any two processors of the same type.

TM-112
Parikh, R.
A DECIDABILITY RESULT FOR A SECOND ORDER PROCESS LOGIC
Pages: 18 September 1978 $3.00
Keywords: dynamic logic, theory of programs, decidability
Abstract: We prove the decidability of the validity problem for a rather general language for talking about computations. As corollaries of our result, we obtain some decidability results of Pratt, Constabel, Fischer-Ladner, and Pnueli and also a new decidability result for deterministic propositional dynamic logic.

TM-113
Pratt, V.R.
A NEAR-OPTIMAL METHOD FOR REASONING ABOUT ACTION
Pages: 24 September 1978 $3.00
Keywords: dynamic logic, program verification, decision method, exponential time, truth set, Hintikka set
Abstract: We give an algorithm for "before-after" reasoning about action. The algorithm decides satisfiability and validity of formulae of propositional dynamic logic, a recently developed logic of change of state that subsumes the zero-order component of most other action-oriented logics. The algorithm requires time at most proportional to an exponentially growing function of the length (number of occurrences of variables and connectives) of the input. Fischer and Ladner have shown that every algorithm for this problem must take exponential time, making this algorithm optimal to within a polynomial. No decision method for any other logic is known to be optimal to within less than an exponential. The typical time for our algorithm makes it a heuristically efficient algorithm of considerable practical interest. Application areas include program verification, program synthesis, and discourse analysis. The algorithm is based on the method of semantic tableaux, appropriately generalized to dynamic logic. A novel treatment of Hintikka sets via theory algebras supplies the theoretical basis for our treatment of tableaux.

TM-114
Dennis, J.B., Fuller, S.H., Ackerman, W.B., Swan, R.J., Weng, K.-S.
RESEARCH DIRECTIONS IN COMPUTER ARCHITECTURE
Pages: 43 September 1978 $3.00
Keywords: computer architecture, computer science research, distributed computer systems, multiprocessor computer systems, supercomputers, computer networks, personal computers

Abstract: The "architecture" of a computer system defines theinterface that the hardware presents to the software of the system, and determines how this interface is realized by subunits of the computer system. In the early days of the stored program computer, when simple "von Neumann" form of main memory and "arithmetic logic unit" was unquestioned, knowledge of logic design, the technology of logic and memory devices, elementary machine language programming techniques, and a good measure of common sense were all that was required to be a computer architect. Now, things have changed. Our concept of what we expect a computer system to do for us has reached a high degree of sophistication — large data bases, multiple concurrent process, and programming languages that offer recursive programming, abstract data types, protection, and access control. These expectations have been met by elaborate software systems — operating systems, data management systems, and runtime support for language implementations. The ability of system designers to meet these expectations, and the quality of the facilities they can provide to the application programmer, are critically dependent on properties of mechanisms built into the hardware. Thus it has become essential that contemporary computer architects be aware of how architectural decisions interact with software quality — how hardware structures can more effectively meet the needs of operating systems and modern concepts of program and data structure. {AD A061-222}

TM-115
Bryant, R.E., Dennis, J.B.
CONCURRENT PROGRAMMING
Pages: 27 October 1978 $3.00
Keywords: concurrency, nondeterminism, semaphores, monitors, message passing, actors, data flow programming
Abstract: Concurrency of activities has long been recognized as an important feature in many computer systems. These systems allow concurrent operations for a number of reasons of whcih three are particularly common. First, by executing several jobs simultaneously, multiprogramming and time-sharing systems can make fuller use of the computing resources. Second, real-time transaction systems, such as airline reservation and point-of-sale terminal systems, allow a number of users to access a single data base concurrently and to obtain responses in real-time. Finally, high speed parallel computers such as array processors dedicate a number of processors to the execution of a single program to speed up completion of computation. {AD A061-180}

TM-116
Pratt, V.R.**APPLICATIONS OF MODAL LOGIC TO PROGRAMMING**
Pages: 25 December 1978 $3.00
Keywords: dynamic logic, logics of programs, predicate calculus, modal logic, referential transparency
Abstract: The model logician's notion of possible world and the computer scientist's notion of state of a machine provide a point of commonality which can form the foundation of a logic of action. Extending ordinary modal logic with the calculus of binary relations leads to a very natural logic for describing the behavior of computer programs.

TM-117
Pratt, V.R.
SIX LECTURES ON DYNAMIC LOGIC
Pages: 23 December 1978 $3.00
Keywords: dynamic logic, logics of programs, semantics of programs, predicate calculus, modal logic, referential transparency

Abstract: The distinction made here between static and dynamic logic has a very simple character, yet can play a central and unifying role in logic as a vantage point from which one can compare propositional calculus, predicate calculus, intentional logics such as modal logic and temporal logic, various algorithmic logics (logics of programs), and Quine's notions of transparency and opacity.

TM-118
Borkin, S.A.
DATA MODEL EQUIVALENCE
Pages: 33 December 1978 $3.00
Keywords: data base systems, semantic graph data model, data model equivalence, semantic relation data model, semantic graph data model
Abstract: The current proliferation of proposals for data base system data models and the desire for data base systems which support several different data models raise many questions concernint "equivalence properties" of different data models. To answer these questions, one first needs clear definitions of the concepts under discussion. This paper presents formal definitions of the terms *data base, operation, operation type, application model* and *data model.*

Using this formal framework, *data base state equivalence, operation equivalence, application model equivalence* and *data model equivalence* are distinguished. Three types of application and data model equivalence are defined — *isomorphic, composed operation* and *state dependent.* Possibilities for *partial equivalences* are mentioned. Implementation implications of these different equivalences are discussed.

Examples are presented using *two semantic data models,* the *semantic relation data model* and the *semantic graph data model.* {AD A062-753}

TM-119
Shamir, A., Zippel, R.E.
ON THE SECURITY OF THE MERKLE-HELLMAN CRYPTOGRAPHIC SCHEME
Pages: 12 December 1978 $3.00
Keywords: cryptography, public-key cryptosystems, Merkle-Hellman cryptosystems, Graham-Shamir knapsacks
Abstract: In this paper we show that a simplified version of the Merkle-Hellman public-key cryptographic system is breakable. While their full-fledged system seems to be resistant to the cryptanalytic attack we propose, this result suggests some ways in which the security of their system can be further enhanced. {AD A063-104}

TM-120
Brock, J.D.
OPERATIONAL SEMANTICS OF A DATA FLOW LANGUAGE
Pages: 55 S.M. Thesis/December 1978 $3.00
Keywords: fixpoint semantics, operational semantics, data flow programming, applicative languages
Abstract: A data flow machine achieves high performance by the concurrent execution of machine code consisting of data flow graphs which explicitly represent the data dependencies among program instructions. This thesis presents the operational semantics of ADFL, an applicative data flow language with an iteration construct resembling tail recursion and an error-handling scheme appropriate to the concurrency of data flow. The operational semantics ***O * T*** of ADFL are expressed by a two step process. The translation algorithm ***T*** maps an ADFL expression into its graph implementation, and the semantics of these graphs are derived by use of Kahn's fixpoint theory of communicating processes. {AD A062-997}

TM-121
Jaffe, J.M.
THE EQUIVALENCE OF R. E. PROGRAMS AND DATA FLOW SCHEMES
Pages: 35 January 1979 $3.00
Keywords: data flow schemes, r.e. programs, effective functionals, Turing machine, computability
Abstract: The expressive power of the data flow schemes of Dennis is evaluated. It is shown that data flow schemes have the power to express an arbitrary determinate functional. The proof involves a demonstration that "restricted data flow schemes" can simulate Turing Machines. This provides a new, simple basis for computability.

TM-122
Jaffe, J.M.
EFFICIENT SCHEDULING OF TASKS WITHOUT FULL USE OF PROCESSOR RESOURCES
Pages: 40 January 1979 $3.00
Keywords: scheduling, list schedules, worst case performance, preemptiveschedules, nonpreemptive schedules
Abstract: The nonpreemptive scheduling of a partially ordered set of tasks on a machine with *m* processors of different speeds is studied. Heuristics are presented which benefit from selective non-use of slow processors. The performance of these heuristics is asymptotic to (sq.rt. *m*) times worse than optimal, whereas demand driven schedules are unboundedly worse than optimal for any fixed value of *m*.

The algorithms are extended to the situation where functionally dedicated processors must process tasks of a given type. Here, too, the worst case performance of the algorithms improves on the worst case performance of known algorithms. The techniques of analyzing these schedules are used to obtain a bound on a large class of preemptive schedules.

TM-123
Perry, H.M.
AN IMPROVED PROOF OF THE RABIN-HARTMANIS-STEARNS CONJECTURE
Pages: 45 S.M. & E.E. Thesis/January 1979 $3.00
Keywords: Rabin-Hartmanis-Sterns conjecture, real time language recognition, Turing machine, overlap argument
Abstract: We offer an improved presentation of Aanderaa's constructive proof of the Rabin-Hartmanis Stearns conjecture:

For all $k \geq 2$, there exists a language L_k such that L_k can be recognized by a k-worktape real time Turing machine but cannot be recognized by any (k-1)-worktape real time Turing machine.

TM-124
Toffoli, T.
BICONTINUOUS EXTENSIONS OF INVERTIBLE COMBINATORIAL FUNCTIONS
Pages: 12 January 1979 $3.00
Keywords: invertible combinatorial functions, continuous extensions, reversible computing, Boolean functions
Abstract: We discuss and solve the problem of constructing a diffeomorphic componentwise extension for an arbitrary invertible combinatorial function. Interpreted in physical terms, our solution constitutes a proof of the physical realizability of general computing mechanisms based on *reversible* primitives. {AD A063-886}

TM-125
Shamir, A., Rivest, R., Adelman, L.M.
MENTAL POKER
Pages: 7 FEBRUARY 1979 $3.00
Keywords: poker, cryptography
Abstract: Is it possible to play a fair game of "Mental Poker?" We will give a complete (but paradoxical) answer to this question. We will first prove that the problem is intrinsically insoluble, and then describe a fair method of playing "Mental Poker." {AD A066-331}

TM-126
Meyer, A.R., Paterson, M.S.
WITH WHAT FREQUENCY ARE APPARENTLY INTRACTABLE PROBLEMS DIFFICULT?
Pages: 9 February 1979 $3.00
Keywords: mathematics, polynomial-time, NP-complete
Abstract: An algorithm is almost polynomial-time (apt) if there is a polynomial *p* such that for all *n*, the algorithm halts within *p*(*n*) inputs of size at most *n*. It is shown that for **NP**-complete and polynomial space-complete problems, as well as certain other apparently intractable problems such as integer factoring, the following conditions are equivalent: (1) the problem is solvable by an apt algorithm, (2) the problem (or its complement) is polynomial-time transformable to a polynomial-sparse set, (3) the problem is solvable in polynomial time.

TM-127
Strazdas, R.J.
A NETWORK TRAFFIC GENERATOR FOR DECNET
Pages: 112 S.B. & S.M. Thesis/March 1979 $3.00
Keywords. networks, DECNET
Abstract. Computer network traffic generators provide a means for supplying benchmark results and for measuring computer network performance at all levels. Eventually they will also aid in fault diagnosis. The network traffic generator described in this thesis allows flexible yet convenient control over a number of parameters useful for generating loads over both test and real networks based on DEC's PDP-11 minicomputer. Implementation on a test network provides sample results. A discussion of design compromises, and recommendations for further study and design point to various open issues.

TM-128
Loui, M C
MINIMUM REGISTER ALLOCATION IS COMPLETE IN POLYNOMIAL SPACE
Pages 23 March 1979 $3.00
Keywords register allocation, pebble game, directed graphs, polynomial space complete, computational complexity
Abstract The Minimum Register Allocation Problem is to determine the number of registers required to evaluate an arithmetic expression. A pebble game on directed acyclic graphs is used to prove that this problem is complete in polynomial space.

TM-129
Shamir, A.
ON THE CRYPTOCOMPLEXITY OF KNAPSACK SYSTEMS
Pages 26 April 1979 $3.00
Keywords cryptography, digital signatures, NP-complete problems, knapsack problems
Abstract: A recent trend in cryptographic systems is to base their encryption/decryption functions on NP-complete problems, and in particular on the knapsack problem. To analyze the security of these systems, we need a complexity theory which is less worst-case oriented and which takes into account the extra conditions imposed on the problems to make them cryptographically useful. In this paper we consider the two classes of one-to-one and onto knapsack systems, analyze the complexity of recognizing them and of solving their instances, introduce a new complexity measure (median complexity), and show that this complexity is inversely proportional to the density of the knapsack system. The tradeoff result is based on a fast probabilistic knapsack solving algorithm which is applicable only to one-to-one systems, and it indicates that knapsack-based cryptographic systems in which one can both encrypt and sign messages are relatively insecure. {AD A067-972}

TM-130
Greif, I., Meyer, A.R.
SPECIFYING THE SEMANTICS OF WHILE-PROGRAMS: A TUTORIAL AND CRITIQUE OF A PAPER BY HOARE AND LAUER
Pages 35 April 1979 $3.00
Keywords: semantics of programming languages
Abstract: We consider three kinds of mathematical objects which can be designated as the "meaning" or "semantics" of programs: binary relations between initial and final states, binary relations on predicates (partial correctness semantics), and functionals from predicates to predicates (predicate transformers). We exhibit various formal specification mechanisms: induction on program syntax, axioms, and deductive systems We show that each kind of semantics can be specified by several different mechanisms. As long as arbitrary predicates on states are permitted, each kind of semantics uniquely determines the others — with the sole exception of the weakest pre-condition semantics for nondeterministic programs. {AD A068-967}

TM-131
Adelman, L.M.
TIME, SPACE AND RANDOMNESS
Pages 19 April 1979 $3.00
Keywords: computational complexity, information theory, NP ^{5} P, factorization
Abstract: Space and time are the fundamental parameters of complexity theory. The thesis of this paper is that randomness is of equal importance. We introduce a notion of randomness (based on Kolomogorov-Chaitin-Randomness), which we suspect will contribute to the understanding of some of the central problems in complexity theory. The purpose of this paper is primarily conceptual, though several easy theorems are given which clarify the relationship of this notion of randomness to the NP^{5}P question, the complexity of integer factoring, and the sets computable in random polynomial time. Finally, using factoring as an exmple, we raise the possibility of performing experiments on functions of unknown complexity to indicate the extent of their tractability

TM-132
Patil, R.S
DESIGN OF A PROGRAM FOR EXPERT DIAGNOSIS OF ACID BASE AND ELECTROLYTE DISTURBANCES
Pages: 39 May 1979 $3.00
Keywords: medical diagnosis, clinical decision making
Abstract: This research develops the diagnostic component of an interactive system for providing expert advice for the diagnosis, therapy and ongoing management of patients with acid-base and electrolyte disturbances We have developed a hierarchic representation of a patients illness which unifies the known facts about the patient, their suspected interrelationships, the hypotheses and how hypotheses account for various known and hypothesized facts. An expectation driven problem solver based on the hypothesize and reformulate paradigm performs the diagnosis.

TM-133
Loui, M.C.
THE SPACE COMPLEXITY OF TWO PEBBLE GAMES ON TREES
Pages: 21 May 1979 $3.00
Keywords: pebble game, computational complexity
Abstract: In the standard pebble game the number of pebbles required to pebble the root of a tree can be computed in time linearly proportional to the number of nodes. For the black/white pebble game the number of pebbles necessary to pebble root of a complete tree is derived.

TM-134
Shamir, A.
HOW TO SHARE A SECRET
Pages: 7 May 1979 $3.00
Keywords: cryptography, key management, interpolation
Abstract: In this paper we show how to divide data D into n pieces in such a way that D is easily reconstructable from any k pieces, but even complete knowledge of k-1 pieces reveals absolutely no information about D. This technique enables the construction of robust key management schemes for cryptographic systems that can function securely and reliably even when misfortunes destroy half the pieces and security breaches expose all but one of the remaining pieces. {AD A069-397}

TM-135
Wyleczuk, R.H.
TIMESTAMPS AND CAPABILITY-BASED PROTECTION IN A DISTRIBUTED COMPUTER FACILITY
Pages: 134 S.B. & S.M. Thesis/June 1979 $3.00
Keywords: distributed computer systems, timestamps, capabilities, protection
Abstract: This thesis investigates the problems of supporting security requirements and providing protection mechanisms in a distributed computer facility. The nature of the environment necessitates examination of operating systems, data base systems, and computer networks. The capability approach to providing protection in a centralized system is chosen as the foundation for the protection mechanism of the distributed system.

The thesis also relies on an interesting approach to the representation of objects in a computer system. An object is represented by a sequence of immutable versions that represent the state of the object over time; each version is the result of an update on the object. This approach to describing objects provides the basis for a flexible definition of the world in which timestamps are naturally associated with every object in the system

The development of a DCF capability mechanism resulted in the following discoveries: Capabilities need not become immediately effective upon their generation. It is not necessary that the object to which access is being authorized exist at the time the capability is generated. And, the revocation of access privileges and the control of capability propagation are not insurmountable problems even in a distributed environment.

TM-136
Misunas, D.P.
REPORT ON THE SECOND WORKSHOP ON DATA FLOW COMPUTER AND PROGRAM ORGANIZATION
Pages: 70 June 1979 $3.00
Keywords: computer architecture, computer systems, programming languages, data flow computation
Abstract: The following report comprises an edited transcription of presentations made at the Second Workshop on Data Flow Computer and Program Organization, held at MIT on July 9-13, 1978, and co-sponsored by the Lawrence Livermore Laboratory (LLL) and the Department of Energy, Mathematical Sciences Branch. These informal transcriptions are only intended to provide a general picture of ongoing work in the area, and to that end, have been heavily edited and often summarized.

TM-137
Davis, E., Jaffe, J.M.
ALGORITHMS FOR SCHEDULING TASKS ON UNRELATED PROCESSORS
Pages: 28 June 1979 $3.00
Keywords: nonpreemptive schedules, worst case finishing time, performance, unrelated processors, largest processing time
Abstract: Several algorithms are presented for the nonpreemptive assignment of *n* independent tasks to *m* unrelated processors. One algorithm requires polynomial time in *n* and *m*, and is at most 2(sq.rt. *m*) times worse than optimal in the worst case. This is the best polynomial time algorithm known for scheduling such sets of tasks. An algorithm with slightly better worst case performance requires polynomial time in *n* but exponential time in *mF0. This is the best algorithm known that requires time 0(n*log(*n*)) for every fixed value of *m*.

TM-138
Pratt, V.R.
DYNAMIC ALGEBRAS: EXAMPLES, CONSTRUCTIONS, APPLICATIONS
Pages: 29 July 1979 $3.00
Keywords: dynamic algebra, logic, program verification, regular algebra
Abstract: Dynamic algebras combine the classes of Boolean (B V ' 0) and regular (R ∪ ; *) algebras into a single finitely axiomatized variety (***B R***) resembling an R-module with "scalar" multiplication The basic result is that * is reflexive transitive closure, contrary to the intuition that this concept should require quantifiers for its definition. Using this result we give several examples of dynamic algebras arising naturally in connection with additive functions, binary relations, state trajectories, languages, and flowcharts. The main result is that free separable dynamic algebras are residually separable-and-finite, important because finite separable dynamic algebras are isomorphic to Kripke structures. Applications include a new completeness proof for the Segerberg axiomatization of propositional dynamic logic, and yet another notion of regular algebra.

TM-139
Martin, W.A.
ROLES, CO-DESCRIPTORS AND THE FORMAL REPRESENTATION OF QUANTIFIED ENGLISH EXPRESSIONS
Pages: 77 May 1980 $3.00
Keywords: knowledge representation, quantifiers, semantic network
Abstract: In representating the semantics of English sentences it is traditional to distinguish logical form from semantic content. The logical form is represented by some sort of predicate calculus. In computational linguistics this predicate calculus or lambda calculus notation is usually carried over, a) directly, b) by replacing parenthesized scope with 'contents' (Hendrix 1978) and, c) by replacing universal quantification with dynamically scoped iteration procedures (Woods 1977).

This paper proposes another possibility for representing logical form. It is based on six main ideas. 1) The use of roles in a semantic net. 2) The referential / attributive distinction. 3) The distributive / collective distinction. 4) The use of two levels of representation for quantified expressions. 5) The use of predicates taking sets, kinds, or prototypical individuals as arguments. 6) The use of meta-description. .br The proposed scheme uses more mechanisms than older ones, but I will argue that it captures subtle cases in a more natural way, and that it is a superior representation for computational use. {AD A074-625}

TM-140
Szolovits, P.
ARTIFICIAL INTELLIGENCE AND CLINICAL PROBLEM SOLVING
Pages: 113 September 1979 $3.00
Keywords: knowledge based systems, artificial intelligence, computer diagnosis, clinical decision making, problem solving
Abstract: An ambitious, but intriguing, possibility for radically increasing the availability and adequacy of health care, while containing its cost, is to use the computer as a consultant to augment and extend the skills of all health care providers. We propose to pursue a program of fundamental research in representation of knowledge, decision making, problem solving, program explanation and clinical cognition, to understand how to construct computer programs that, as an integral part of the health care system, can amplify the knowledge and reasoning powers of medical decision makers. We plan to apply the techniques so developed, to problems in the diagnosis and therapy of acid/base and electrolyte disturbances, diagnosis of birth defects using an existing data base of diseases and associated manifestations, the development of multi-modal cancer therapy protocols, and the application of the methods of decision analysis to produce general tools for physicians to use in analyzing difficult clinical cases.

TM-141
Hammer, M.M., McLeod, D.
ON DATA BASE MANAGEMENT SYSTEM ARCHITECTURE
Pages: 39 October 1979 $3.00
Keywords: data base management, conceptual data models, data base semantics
Abstract: Despite the many advances that have been made in the field of data base management in the last two decades, in many respects the paradigm of data base management has not changed much since its inception. Several long-standing assumptions pervade the field and exert a great influence on the architecture of data base management systems, their functions, and the kinds of data bases that they manage. This paper reconsiders some of these assumptions and suggests certain alternatives to them. In particular, it is argued that the concept of an *integrated data base* ought to be supplanted by that of a *federated data base*, a loose assembly of semi-independent components; the position of the data base management system in the context of a total information system is reexamined, and arguments are made for extending its functional capabilities; and *controlled logical redundancy* in the schema is introduced as a means of improving the usability of a data base and of enhancing its life-cycle performance. An underlying theme throughout is that of the importance of a *semantic schema* of the data base, which specifies enough of the meaning of the application domain to enable enhanced functionality to be achieved. A number of characteristics of a conceptual data model (in which this schema would be expressed) are described. {*AD A076-417*}

TM-142
Lipski, W.
ON DATA BASES WITH INCOMPLETE INFORMATION
Pages: 64 October 1979 $3.00
Keywords: data base, incomplete information, query language semantics, implicit information, modal logic, relational model, null values
Abstract: Semantic and logical problems arising in an incomplete information data base are investigated. A simple query language is described, and its semantics is defined, which refers the queries to the information about reality contained in a data base, rather than to reality itself. This approach, called the internal interpretation, is shown to lead in a natural way to the notions of a topological Boolean algebra and a modal logic related to S4, in the same way as referring queries directly to reality (external interpretation) leads to Boolean algebras and classical logic. An axiom system is given for equivalent (with respect to the internal interpretation) transformation of queries, which is then exploited as a basic tool in a method for computing the internal interpretation for a broad class of queries. An interesting special case of the problem of determining the internal interpretation amounts to deciding whether an assertion about reality (a "yes-no" query) is consistent with the incomplete information about reality contained in a data base. We give a solution to this problem, which relies on the classic combinatorial problem of distinct representatives of subsets.

TM-143
Leth, J. W.
AN INTERMEDIATE FORM FOR DATA FLOW PROGRAMS
Pages: 117 S.M Thesis/November 1979 $3.00
Keywords. data flow, VAL, applicative languages, parallel programming, graph representation
Abstract: A data flow program, often represented as a data flow graph, is a program that expresses a computation by indicating the data dependencies among operators. A data flow computer is a machine designed to take advantage of *concurrency* in data flow graphs by executing data-independent operations in parallel (that is, a sequential ordering exists only between operations for which the result of one operation is an operand of the other). This thesis presents a form of computer representation of data flow programs (based on data flow graphs) that can serve as an intermediate form in the translation of source language code into machine code for a data flow computer. The proposed intermediate representation is implemented in the structured programming language CLU, and is designed to allow analysis and transformation of programs (for optimization purposes) to be performed either automatically or with programmer interaction.

TM-144
Takagi, A.
CONCURRENT AND RELIABLE UPDATES OF DISTRIBUTED DATA BASE
Pages: 70 November 1979 $3.00
Keywords: distributed data base, concurrency control, recovery, recoverable objects, atomicity, multiple versions
Abstract: A concurrent execution of transactions and various failures occuring during transaction processing in a distributed data base system can lead to an inconsistent data base state. In order to prevent such inconsistency form occuring, 1) the schedule of transactions must be equivalent to some serial schedule and 2) each transaction must be either completed or backed out. This paper develops a set of schemes that satisfy these requirements and still realize highly concurrent execution of transactions. This paper also shows how to incorporate these schemes into a multi-level distributed data base system where there exists a hierarchy of transactions. Detailed algorithms for concurrent and reliable updates of distributed data bases based on the proposed schemes are included in the appendix.

TM-145
Loui, M.C.
A SPACE BOUND FOR ONE-TAPE MULTIDIMENSIONAL TURING MACHINES
Pages: 15 November 1979 $3.00
Keywords: multidimensional Turing machine, Turing machine, time-space tradeoff, time complexity, tape complexity, computational complexity
Abstract: Let L be a language recognized by a nondeterminsitic Turing machine with one d-dimensional worktape of time complexity $T(n)$. Then L can be recognized by a deterministic Turing machine of space complexity $(T(n) \log T(2n))^{d/d+1}$. The proof employs a generalized crossing sequence argument.

TM-146
Aoki, D.J.
A MACHINE LANGUAGE INSTRUCTION SET FOR A DATA FLOW PROCESSOR
Pages: 63 S.M. Thesis/December 1979 $3.00
Keywords: data flow computer, program graphs, data flow instruction cell, safety, instruction cells, instruction set
Abstract: A data flow processor is a computer in which instructions are data driven and enabled for execution by the arrival of their operands. Data flow processors execute data flow programs, normally represented as program graphs, which represent the data dependencies between operations. This thesis presents a machine language instruction set for a Form 1 data flow machine based on the Dennis-Misunas design.

TM-147
Schroeppel, R., Shamir, A.
A $T = 0(2^{n/2})$, $S = 0(2^{n/4})$ ALGORITHM FOR CERTAIN NP-COMPLETE PROBLEMS
Pages: 20 January 1980 $3.00
Keywords: NP-complete, time/space tradeoffs, knapsack problems, Merkle-Hellman cryptosystems
Abstract: n this paper we develop a general purpose algorithm that can solve a number of NP-complete problems in time $T=0(2^{n/2})$and space $S=0(2^{n/4})$. The algorithm can be generalized to a family of algorithms whose time and space complexities are related by $T*S^2=0(2^n)$. The problems it can handle are characterized by a few decomposition axioms, and they include knapsack problems, exact satisfiability problems, set covering problems, etc. The new algorithm has a considerable cryptanalytic significance, since it can break knapsack-based cryptosystems with up to n = 100 generators. {AD A080-385}

TM-148
Adelman, L.M., Loui, M.C.
SPACE-BOUNDED SIMULATION OF MULTITAPE TURING MACHINES
Pages: 12 January 1980 $3 00
Keywords: Turing machine, time complexity, space complexity, overlap
Abstract: A new proof of a theorem of Hopcroft, Paul and Valiant is presented: every deterministic multitape Turing machine of time complexity T(n) can be simulated by a deterministic Turing machine of space complexity T(n)/log T(n). The proof includes an overlap argument.

TM-149
Pallottino, S., Toffoli, T.
AN EFFICIENT ALGORITHM FOR DETERMINING THE LENGTH OF THE LONGEST DEAD PATH IN AN "LIFO" BRANCH-AND-BOUND EXPLORATION SCHEMA
Pages: 9 January 1980 $3 00
Keywords: life of longest dead path, branch-and-bound, LIFO tree search
Abstract: The length of the longest dead path (LLDP) is a widely used parameter in estimating the efficiency of branch-and-bound optimization algorithms that employ the LIFO exploration schema. Thanks to two original theorems, we are able to present a particularly attractive procedure for determining of the LLPD. In fact, this procedure requires a number of storage variables which is independent of problem size and very small; moreover, the procedure is self-contained in the sense that it can be externally attached to any LIFO branch-and-bound program without interfering with its algorithms and data structures. {AD A079-912}

TM-150
Meyer, A.R.
TEN THOUSAND AND ONE LOGICS OF PROGRAMMING
Pages: 19 February 1980 $3.00
Keywords: logics of programs
Abstract: A summary of two years of Professor Meters work and the work of his colleagues at M.I.T. on logics of programs.

TM-151
Toffoli, T.
REVERSIBLE COMPUTING
Pages: 36 February 1980 $3.00
Keywords: reversible computing, computation universality, automata, networks, physical computing
Abstract: The theory of reversible computing is based on invertible primitives and composition rules that preserve invertibility. With these constraints, one can still satisfactorily deal with both functional and structural aspects of computing processes; at the same time, one attains a closer correspondence between the behavior of abstract computing systems and the microscopic physical laws (which are presumed to be strictly reversible) that underly any concrete implementation of such systems.

Here, we integrate into a comprehensive picture a variety of concepts and results. According to a physical interpretation, the central result of this paper is that it is ideally possible to build sequential circuits with zero power dissipation. Even when these circuits are interfaced with conventional ones, power dissipation at the interface would be at most proportional to the number of input/output lines, rather than to the number of logic gates as in conventional computers. {AD A082-021}

TM-152
Papadimitriou, C.H.
ON THE COMPLEXITY OF INTEGER PROGRAMMING
Pages: 6 February 1980 $3 00
Keywords: integer linear programming, P and NP, pseudopolynomial algorithms
Abstract: We give a simple proof that integer programming is in NP. Our proof also establishes that there is a pseudopolynomial time algorithm for integer programming with any (fixed) number of constraints.

TM-153
Papadimitriou, C.H.
WORST-CASE AND PROBABILISTIC ANALYSIS OF A GEOMETRIC LOCATION PROBLEM
Pages: 21 February 1980 $3.00
Keywords: location problems, K-median problem, NP-complete, probabilistic algorithms
Abstract: We consider the problem of choosing K "medians" among *n* points on the Euclidean plane such that the sum of the distances from each of the *n* points to this closest median is minimized. We show that this problem is NP-complete. We also present two heuristics that produce arbitrarily good solutions with probability going to one. One is a partition heuristic, and works when K grows linearly —or almost so— with *n*. The other is the "honeycomb" heuristic, and is applicable to rates of growth of K of the form K $n\in$, $0<\in<1$.

TM-154
Karp, R.M., Papadimitriou, C.H.
ON LINEAR CHARACTERIZATIONS OF COMBINATORIAL OPTIMIZATION PROBLEMS
Pages: 17 February 1980 $3.00
Kewords: combinatorial optimization, linear programming, convex polytopes, P, NP, and co-NP, Khacians algorithm
Abstract: We show that there can be no computationally tractable description by linear inequalities of the polyhedron associated with any NP-complete combinatorial optimization problem unless NP = co-NP — a very unlikely event. We also use the recent result by Khachian to present even stronger evidence that NP-complete combinatorial optimization problems cannot have efficient generators of violated inequalities.

TM-155
Itai, A., Lipton, R.J., Papadimitriou, C.H., Rodeh, M.
COVERING GRAPHS BY SIMPLE CIRCUITS
Pages: 7 February 1980 $3.00
Keywords: graph algorithms, Eulerian subgraphs, edge connectivity
Abstract: A family $C_1,...,C_m$ of simple circuits of an undirected multigraph G ^{5}(V,E) covers G provided each edge of G is in one of the circuits. The size of such a family is then the sum of the lengths of the circuits $C_1,...,C_m$. We are interested here in the question of finding covers of minimum size. Clear, we can restrict our attention to 2-connected multigraphs: if a graph has a bridge then it has no cover at all.

TM-156
Meyer, A.R., Parikh, R.
DEFINABILITY IN DYNAMIC LOGIC
Pages: 15 February 1980 $3.00
Keywords: dynamic logic, infinitary logic
Abstract: We study the expressive power of various versions of Dynamic Logic and compare them with each other as well as with standard languages in the logical literature. One version of Dynamic Logic is equivalent to the infinitary logic $L^{CK}_{\omega 1 \omega}$, but regular Dynamic Logic is strictly less expressive.

In particular, the ordinals ω^{ω} and $\omega^{\omega}*2$ are indistinguishable by formulas of regular Dynamic Logic.

TM-157
Meyer, A.R., Winklmann, K.
ON THE EXPRESSIVE POWER OF DYNAMIC LOGIC
Pages: 36 February 1980 $3.00
Keywords: looping, dynamic logic
Abstract: We show that "looping" of while-programs can be expressed in Regular First Order Dynamic Logic, disproving a conjecture made by Harel and Pratt. In addition we show that the expressive power of quantifier-free Dynamic Logic increases when nondeterminism is introduced in the programs that are part of formulae of Dynamic Logic. Allowing assignments of random values to variables also increases expressive power.

TM-158
Stark, E.W.
SEMAPHORE PRIMITIVES AND STARVATION-FREE MUTUAL EXCLUSION
Pages: 167 S.M. Thesis/March 1980 $3.00
Keywords: parallel processing, mutual exclusion, semaphores, synchronization
Abstract: Most discussions of semaphore primitives in the literature provide only an informal description of their behavior, rather than a more precise definition. These informal descriptions may be incorrect, incomplete, or subject to misinterpretation. As a result, the literature actually contains several different definitions of the semaphore primitives. The differences are important, since the particular choice of semaphore primitives allows the possibility of process *starvation*. This thesis attempts to alleviate some of the confusion by giving precise definitions of two varieties of semaphore primitives; here called weak and blocked-set primitives. It is then shown that under certain natural conditions, although it is possible to implement starvation-free mutual exclusion with blocked-set semaphores, it is not possible to do so with weak semaphores. Thus weak semaphores are strictly less "powerful" than blocked-set semaphores.

TM-159
Pratt, V.R.
DYNAMIC ALGEBRAS AND THE NATURE OF INDUCTION
Pages: 15 March 1980 $3.00
Keywords: dynamic algebra, logic, program verification, regular algebra,Segerberg axioms
Abstract: Dynamic algebras constitute the variety (equationally defined class) of models of the Segerberg axioms for propositional dynamic logic. We obtain the following results (to within inseparability). (i) In any dynamic algebra * is reflexive transitive closure. (ii) Every free dynamic algebra can be factored into finite dynamic algebras. (iii) Every finite dynamic algebra is isomorphic to a Kripke structure. (ii) and (iii) imply Parikh's completeness theorem for the Segerberg axioms. We also present an approach to treating the inductive aspect of recursion within dynamic algebras.

TM-160
Kanellakis, P.C.
ON THE COMPUTATIONAL COMPLEXITY OF CARDINALITY CONSTRAINTS IN RELATIONAL DATA BASES
Pages: 7 March 1980 $3.00
Keywords: computational complexity, NP-complete, relational data base, cardinality constraints
Abstract: We show that the problem of determining whether or not a lossless join property holds for a data base, in the presence of key dependencies and cardinality constraints on the domains of the attributes is NP-complete.

TM-161
Lloyd, E.L.
CRITICAL PATH SCHEDULING OF TASK SYSTEMS WITH RESOURCE AND PROCESSOR CONSTRAINTS
Pages: 34 March 1980 $3.00
Keywords: scheduling, task system with resources, bin packing
Abstract: Several papers over the past few years have investigated minimum execution time scheduling of unit execution time (UET) task systems with resources. Because such scheduling problems are, in general, NP-hard, a variety of heuristic methods for producing schedules have been studied, among them, critical path scheduling. The strongest results to date have been for systems where there is no processor constraint. These results may be utilized for system with a processor constraint by treating the processors as an additional resource. Unfortunately, in those cases where the number of processors is close to the number of resources, this results in an upper bound which is somewhat misleading. In this paper we investigate the performance of critical path scheduling for UET task systems with resources and a fixed number of processors. An upper bound for the worst case performance of critical path scheduling is given. This bound depends both on the number of processors and on the number of different resources. Moreover, we show that this is the best possible (asymptotic) upper bound.

TM-162
Marcum, A.M.
A MANAGER FOR NAMED, PERMANENT OBJECTS
Pages: 135 S.B. & S.M. Thesis/April 1980 $3.00
Keywords: file systems, data abstractions, permanent storage
Abstract: Storing data in a computing system for a long time has been of interest ever since it was possible to do so. Classically, one stores bit- or byte-strings, or perhaps arrays of "records." Yet, current programming philosophy stresses data abstraction techniques and concepts.

This report describes an object-oriented filing system which stores abstract objects, and allows the user to view the system as though one were storing abstract objects, rather than storing some external representation of the abstractions. Names may be attached to the (permanent) objects, and objects may be contained in (and may contain) other objects. Furthermore, an object may be contained in more than one object, thereby allowing the naming structure to be a network. {AD A083-491}

TM-163
Meyer, A.R., Halpern, J.Y.
AXIOMATIC DEFINITIONS OF PROGRAMMING LANGUAGES: A THEORETICAL ASSESSMENT
Pages: 34 April 1980 $3.00
Keywords: semantics of programming languages, partial correctness, termination assertions, axiomatic definitions of programming languages
Abstract: A precise definition is given of how partial correctness or termination assertions serve to define the semantics of classes of program schemes. Assertions involving only formulas of first order predicate calculus are proved capable of defining program scheme semantics, and effective axiom systems for deriving such assertions are described. Such axiomatic definitions are possible despite the limited expressive power of predicate calculus.

TM-164
Shamir, A.
THE CRYPTOGRAPHIC SECURITY OF COMPACT KNAPSACKS-PRELIMINARY REPORT
Pages: 20 April 1980 $3.00
Keywords. cryptography, knapsack problems, Merkle-Hellman cryptosystems
Abstract: In 1978, Merkle and Hellman introduced a knapsack-based public-key cryptosystem, which received widespread attention. The two major open problems concerning this cryptosystem are:

(i) Security. How difficult are the Merkle-Hellman knapsacks?

(ii) Efficiency: Can the huge key size be reduced?

In this paper we analyze the cryptographic security of knapsack problems with small keys, develop a new (non-enumerative) type of algorithm for solving them, and use the algorithm to show that under certain assumptions it is as difficult to find the hidden trapdoors in Merkle-Hellman knapsacks as it is to solve knapsack problems. {AD A084-456}

TM-165
Finseth, C.A.
THEORY AND PRACTICE OF TEXT EDITORS OR A COOKBOOK FOR AN EMACS
Pages: 106 S.B. Thesis/May 1980 $3.00
Keywords: redisplay, text editing, video terminals
Abstract: A comprehensive summary of the available technology for implementing text editors. It is written to be a guide for the implementor of a text editor. It does not provide a finished, polished algorithm for any part of a text editor. Rather, it provides a breakdown of the problems involved and discusses the pitfalls and the available tradeoffs to be considered when designing a text editor. Specific reference is made to the relevant tradeoffs for an Emacs-type editor, a character-oriented, extensible display editor.

TM-166
Bryant, R.E.
REPORT ON THE WORKSHOP ON SELF-TIMED SYSTEMS
Pages: 21 May 1980 $3.00
Keywords self-timed systems, logic design, digital systems, asynchronous systems, speed independent, workshop report
Abstract: This workshop, held in July of 1979, served to bring together experts in the field of self-timed systems to review and assess the state of the art and to chart directions for future research For the purpose of the workshop, self-timed systems were defined to include any system composed of a set of modules which communicate asynchronously The modules, however, may themselves by implemented either synchronously or asynchronously It is hoped that the advent of custom LSI parts offers a new opportunity for the application of self-timed principles. The strong bias of conventional, standard IC parts toward clocked systems may no longer limit the practicality of self-timed designs.

TM-167
Pavelle, R., Wester, M.
COMPUTER PROGRAMS FOR RESEARCH IN GRAVITATION AND DIFFERENTIAL GEOMETRY
Pages: 41 June 1980 $3.00
Keywords: MACSYMA, algebraic manipulation, general relativity, gravitation, tensor analysis, differential geometry
Abstract: This report contains a description of all current functions (with many examples) of the programs CTENSR and ITENSR which are available with MACSYMA. CTENSR is a standard Component TENSoR manipulation system which means that gemetrical tensor objects are represented as arrays or matrices. Tensor operations such as contraction or covariant differentiation are carried out by actually summing over repeated (dummy) indices with DO statements. ITENSR, is a unique Indicial TENSoR manipulation system which is implemented by representing tensors as functions of their covariant, contravariant and derivative indices. Tensor operations such as contraction or covariant differentiation are performed by manipulating the indices themselves rather than the components to which they correspond. The programs are connected in the sense that one can obtain an expression in ITENSR and have the corresponding expression generated in the CTENSR format automatically

TM-168
Greif, I.
PROGRAMS FOR DISTRIBUTED COMPUTING: THE CALENDAR APPLICATION
Pages: 6 July 1980 $3.00
Keywords: distributed computing, office automation, calendar management
Abstract: The calendar application involves a wide range of issues in distributed computing, from implementation of distributed data bases to design of a user interface that will enable the user to comprehend the complex distributed environment in which he is working. This memo summarizes current status of design and implementation of calendars. {AD A087-357}

TM-169
Burke, G., Moon, D.
LOOP ITERATION MACRO
Pages: 25 January 1981 $3.00
Keywords: iteration, LISP, Macro
Abstract: LOOP is a Lisp macro which provides a programmable iteration facility. The same LOOP module operates compatibly in both Lisp Machine Lisp and Maclisp (PDP-10 and Multics), and NIL, and a moderately compatible package is under development for the MDL programming environment. LOOP was inspired by the "FOR" facility and CLISP in Interlisp; however, it is not compatible and differs in several details {AD A087-372}

TM-170
Ehrenfeucht, A., Parikh, R., Rozenberg, G.
PUMPING LEMMAS FOR REGULAR SETS
Pages: 11 August 1980 $3.00
Keywords: pumping lemmas, iteration theorems
Abstract: It is well known that regular languages satisfy certain conditions known as pumping lemmas or iteration theorems. However, the question of the converse result has been open. We show that the usual form of the pumping lemma falls very far short of implying regularity, but that there is a form, which we have called the *block pumping property*, that is equivalent to regularity.

TM-171
Meyer, A.R.
WHAT IS A MODEL OF THE LAMBDA CALCULUS?
Pages: 20 August 1980 $3.00
Keywords: lambda calculus, mathematics
Abstract: An elementary, purely algebraic definition of model for the pure, untyped lambda calculus is given. This definition is shown to be equivalent to the usual syntactic definition. A simple construction of models for α-β($-\eta$)-calculus is reviewed.

TM-172
Paseman, W.G.
SOME NEW METHODS OF MUSIC SYNTHESIS
Pages: 110 S.M. Thesis/August 1980 $3.00
Keywords: artificial intelligence, music synthesis
Abstract: The first section discusses music composition, shows why it is a useful domain for Artificial Intelligence research and presents a set of "Design Rules" that facilitate research in the field of tonal music composition. The second section describes some of the problems and issues encountered while designing the initial hardware for the Music Aided Cognition Project at MIT. All of the developed hardware permits computer control, performance and recording of music in real time. {AD A090-130}

TM-173
Hawkinson, L.B.
XLMS: A LINGUISTIC MEMORY SYSTEM
Pages: 49 September 1980 $3.00
Keywords: knowledge representation, semantic network, linguistic memory
Abstract: LMS (**Linguistic Memory System**) is a knowledge representation formalism particularly designed for representing knowledge that can be straightforwardly expressed in natural language. Fundamentally, it is a semantic network formalism, a formalism for managing interconnected objects in a highly-organized, network-like memory. XLMS is a particular LISP-based implementation of LMS, intended primarily for experimental use. {AD A090-033}

TM-174
Arvind, Kathail, V., Pingali, K.
A DATA FLOW ARCHITECTURE WITH TAGGED TOKENS
Pages: 24 September 1980 $3.00
Keywords: data flow, multiple processor architectures, parallel processing, applicative languages, high-performance computers, functional languages, program decomposition, token labeling
Abstract: A machine comprising of hundreds of processing elements must have a highly distributed and asynchronous control structure. We are designing a system based on data flow principles in which each processing element contains a part of the program, and processors communicate by sending information packets to each other. Our machine is a hardware realization of a novel way of interpreting data flow languages known as the U-interpreter. The design of a processing element (PE) and a communication system for such a machine is presented .We also present arguments as to why our architecture can tolerate long average delays in the communication network without affecting the overall performance. Schemes for mapping programs onto this machine are also discussed briefly.

TM-175
Meyer, A.R., Weise, D., Loui, M.C.
ON TIME VERSUS SPACE III
Pages: 8 September 1980 $3.00
Keywords: pebble game, time/space tradeoffs, storage modification machines, time complexity, space complexity
Abstract: Paul and Reishuk devised space efficient simulations of logarithmic cost random access machines and multidimensional Turing machines. We simplify their general space reduction technique and extend it to other models of computation, particularly to the class of storage modification machines (SMM), a model of list processing. Every SMM of time complexity t(n) can be simulated by an SMM of space complexity *t(n)/log t(n).*

TM-176
Seaquist, C.R.
A SEMANTICS OF SYNCHRONIZATION
Pages: 111 S.M. Thesis/September 1980 $3.00
Keywords: synchronization, concurrency
Abstract: This paper presents a rigorous framework in which to discuss the synchronization necessary to coordinate accesses to a resource. The framework, among other things, provides a method for specifying concurrency and forms the semantic basis of a synchronization mechanism which avoids certain unfortunate characteristics of monitors and serializers. Synchronization is viewed as being managed by a resource guardian. A synchronization problem is defined as a predicate on event sequences. The interaction of a guardian and the rest of the system is formalized in terms of a two person game. This formalization results in precise definitions of guardian and guardian behavior. The notion of a "good" or optimal solution is defined, and the solutions to certain classes of synchronization problems are characterized. An abstract description of the general actions of a guardian is given. This general description, with some restrictions, forms the basis of a simple synchronization mechanism for actually implementing solutions. The mechanism is given a rigorous semantics based on the definition of guardian. This facilitates the verification of correctness. Many examples of the use of the mechanism are given and its advantages are discussed. {AD A091-015}

TM-177
Sinha, M.K.
TIMEPAD - A PERFORMANCE IMPROVING SYNCHRONIZATION MECHANISM FOR DISTRIBUTED SYSTEMS
Pages: 55 September 1980 $3.00
Keywords: distributed data base, synchronization, atomicity, performance, timestamps
Abstract: A new mechanism for the synchronization of accesses to distributed data objects is developed. This mechanism, called timepad, is an extension to the timestamp synchronization scheme and it encaches the concurrency transparency requirement of the user reducing the chance of eventual rejection of a transaction. The timepad scheme will improve the performance of those distributed data base systems where the probability of transactions clashing is high.

The timepad scheme is very nicely integrated with the object history concept developed by David Reed and the improved power of the integrated scheme is shown to solve the meeting fixing problem. We also discuss how the timepad mechanism can be used as an approximation pad to solve problems which need approximate solution.

TM-178
Arvind, Thomas, R.E.
I-STRUCTURES: AN EFFICIENT DATA TYPE FOR FUNCTIONAL LANGUAGES
Pages: 19 October 1981 $3.00
Keywords: applicative languages, asynchronous systems, data flow, data structures, data types, functional semantics, multiple processor systems, parallelism, storage management
Abstract· Data structure operations in purely functional languages often consume relatively large amounts of memory and processing time. In many circumstances, however, the full generality of these data structure operations is not needed and hence significant gains should be possible by substituting restricted data structure operations. An I-structure is a new array-like data structure which can substantially reduce data structure overhead in functional programs when data structures are generated or consumed "monotonically". I-structures retain functional semantics thereby aiding the detection and exploitation of parallelism and are especially useful in systems based on data flow principles to reduce data dependencies from an entire data structure to individual elements of the structure. In many cases the reduction of such data dependencies can result in significant gains in parallelism. Unlike other proposals for reducing data dependencies, I-structures are convenient for expressing numerical algorithms since they allow random access to individual elements without additional overhead.

TM-179
Halpern, J.Y., Meyer, A.R.
AXIOMATIC DEFINITIONS OF PROGRAMMING LANGUAGES, II
Pages: 18 October 1980 $3.00
Keywords: semantics, programming languages, partial correctness, axiomatic definitions of programming languages, expressiveness, termination
Abstract: Sufficient conditions are given for partial correctness assertions to determine the input-output semantics of quite general classes of programming languages. This determination cannot be unique unless states which are indistinguishable by predicates in the assertions are identified. Even when indistinguishable states are identified, partial correctness assertions may not suffice to determine program semantics.

TM-180
Papadimitriou, C.H.
A THEOREM IN DATA BASE CONCURRENCY CONTROL
Pages: 6 October 1980 $3.00
Keywords: concurrency, concurrency control, data base transactions, locking, semaphores

Abstract: Consider two straight-line problems A and B, and let H be a set of sequences of steps of A and B, possibly interleaved, but each containing all steps of A and B in the right order. We give a necessary and sufficient condition for H to be realizable as the set of all sequences of steps of A and B. This condition captures the intuitive limitations of locking primitives owing to their limited memory capacity.

TM-181
Lipski, W., Papadimitriou, C.H.
A FAST ALGORITHM FOR TESTING FOR SAFETY AND DETECTING DEADLOCKS IN LOCKED TRANSACTION SYSTEMS
Pages 14 October 1980 $3.00
Keywords. data base concurrency control, geometry of locking, computational geometry
Abstract. We represent an 0(nlognlognloglogn) time algorithm which, given a set of n rectangles on the plane with horizontal and vertical sides, and two points s and t, determines whether there exists a monotonically increasing curve form s to t which separates two of the rectangles while avoiding all other rectangles. This solves several problems related to data base concurrency control

TM-182
Itai, A, Papadimitriou, C.H., Szwarefiter, J.L.
HAMILTON PATHS IN GRID GRAPHS
Pages: 14 October 1980 $3.00
Keywords: Hamilton circuit, grid graphs, rectangular grid graphs, NP-complete, traveling salesman problem
Abstract: A grid path is a node-induced finite subgraph of the infinite grid. It is rectangular if its set of nodes is the product of two intervals. Given a rectangular grid graph and two of its nodes, we give necessary and sufficient conditions for the graph to have a Hamilton path between these two nodes. In contrast, the Hamilton path (and circuit) problem for general grid graphs is shown to be NP-complete. This provides a new, relatively simple, proof of the result that the Euclidean traveling salesman problem is NP-complete.

TM-183
Meyer, A.R.
A NOTE ON THE LENGTH OF CRAIG'S INTERPOLANTS
Pages: 4 October 1980 $3.00
Keywords Craigs interpolants
Abstract: There is no recursive bound on the length of the smallest interpolant.

TM-184
Lieberman, H., Hewitt, C
A REAL TIME GARBAGE COLLECTOR THAT CAN RECOVER TEMPORARY STORAGE QUICKLY
Pages: 22 October 1980 $3.00
Keywords: garbage collection, storage, recovery
Abstract: In previous heap storage systems, the cost of creating objects and garbage collection is independent of the lifetime of the object. Since temporary objects account for a large portion of storage use, it's worth optimizing a garbage collector to reclaim temporary storage faster. We present a garbage collection algorithm which:

(1) Makes short term storage cheaper than long term storage.
(2) Operates in real time-object creation and access times are bounded.
(3) Works well with multiple processors and a large address space.

TM-185
Kung, H -T., Papadimitriou, C.H.
AN OPTIMALITY THEORY OF CONCURRENCY CONTROL FOR DATABASES
Pages: 13 November 1980 $3.00
Keywords: data base concurrency control, scheduling, optimal scheduler
Abstract: A concurrency control mechanism (or a scheduler) is the component of a data base system that feguards the consistency of the data base in the presence of interleaved accesses and update requests. We formally show that the performance of a scheduler, i.e., the amount of parallelism that it supports, depends explicitly upon the amount of information that is available to the scheduler. We point out that most previous work on concurrency control is simply concerned with specific points of this basic trade-off between performance and information. In fact, several of these approaches are shown to be optimal for the amount of information that they use. {AD A092-625}

TM-186
Szolovits, P., Martin, W.A.
BRAND X MANUAL
Pages: 21 November 1980 $3.00
Keywords: LISP, artificial intelligence, knowledge representation, semantic network, programming languages
Abstract: BRAND X is a representation language implemented as a pure extension of LISP. BRAND X provides the following additional facilities over LISP: *Unique* and *canonical* structures, *property lists* for all objects, *labels* for all objects, and a syntax to express each of these, supported by a reader and printer. BRAND X is intended as an "assembly language" for representation languages, attempting to provide facilities generally found useful in the simplest manner, without any strong commitment to specific representational conventions. {AD A093-041/2}

TM-187
Fischer, M.J., Meyer, A.R., Paterson, M.S.
Ω (n log n) **LOWER BOUNDS ON LENGTH OF BOOLEAN FORMULAS**
Pages 18 November 1980 $3.00
Keywords: mathematics, Boolean functions
Abstract: A property of Boolean functions of n variables is described and shown to imply lower bounds as large as Ω(n log n) on the number of literals in any Boolean formula for any function with the property. Formulas over the full basis of binary operations (Λ, $\oplus$, etc) are considered The lower bounds apply to all but a vanishing fraction of symmetric functions, in particular to all threshold functions with sufficiently large threshold and to the "congruent to zero modulo k" function for $k>2$. In the case $k = 4$ the bound is optimal.

TM-188
Mayr, E.W.
AN EFFECTIVE REPRESENTATION OF THE REACHABILITY SET OF PERSISTENT PETRI NETS
Pages: 17 January 1981 $3.00
Keywords: vector replacement system, Petri nets, persistence, representation of reachability set
Abstract: In a persistent net, an enabled transition can become disabled only by firing itself. Here, an algorithm is presented which constructs a semilinear representation of the set of states reachable in an arbitrary persistent Petri net

TM-189
Mayr, E.W.
PERSISTENCE OF VECTOR REPLACEMENT SYSTEMS IS DECIDABLE
Pages: 18 January 1981 $3.00
Keywords: vector replacement system, Petri nets, persistence, representation of reachability set
Abstract: In a persistent vector replacement system (VRS) or Petri net, an enabled transition can become disabled only by firing itself. Here, an algorithm is presented which allows to decide whether an arbitrary VRS is persistent or not, and if so, to construct a semilinear representation of the set of states reachable in the system.

TM-190
Ben-Ari, M., Halpern, J.Y., Pnueli, A.
DETERMINISTIC PROPOSITIONAL DYNAMIC LOGIC: FINITE MODELS, COMPLEXITY, AND COMPLETENESS
Pages: 20 January 1981 $3.00
Keywords: deterministic propositional dynamic logic, propositional dynamic logic, tableau method, completeness
Abstract: Let p be a formula in deterministic propositional dynamic logic. A decision procedure for the satisfiability of p is given along with a construction of finite model for every satisfiable p. The decision procedure runs in deterministic time 2^{cn} and the size of the model is bounded by n^2*4n, where n is the length of p. Finally, a complete axiomatization for deterministic propositional dynamic logic is given, based on the Segerberg axioms for propositional dynamic logic.

TM-191
Parikh, R.
PROPOSITIONAL DYNAMIC LOGICS OF PROGRAMS: A SURVEY
Pages: 37 January 1981 $3.00
Keywords: propositional dynamic logic
Abstract: The use of logic in program verification is an old idea, as such ideas go. Early work by Engeler, Floyd, Hoare and Salwicki has already developed into a rich field with many workers. However the *propositional*

versions of these logics are relatively new; work in this field goes back only to Fischer and Ladner's 1977 paper where they showed that the propositional version of Pratt's dynamic logic is decidable. Since then the field has developed rapidly, a very large number of preliminary questions have already been settled, and interesting side areas are developing. In what follows, we shall try to give an overview, stating some of the main results and referring the reader to the original sources for the more difficult or elaborate proofs.

TM-192
Meyer, A.R., Streett, R.S., Mirkowska, G.
THE DEDUCIBILITY PROBLEM IN PROPOSITIONAL DYNAMIC LOGIC
Pages: 16 February 1981 $3.00
Keywords: propositional dynamic logic, dynamic logic
Abstract: The problem of whether an arbitrary formula of Propositional Dynamic Logic (PDL) is deducible from a fixed axiom scheme of *PDL* is Π_1^1-complete. This contrasts with the decidability of the problem when the axiom scheme is replaced by any single PDL formula.

TM-193
Yannakakis, M, Papadimitriou, C.H.
ALGEBRAIC DEPENDENCIES
Pages: 32 February 1981 $3.00
Keywords· relational data base, data dependencies, template and algebraic dependencies, embedded implicational dependencies, tableaux, extended relations, complete axiomatization, project-join expressions
Abstract: We propose a new kind of data dependencies called algebraic dependencies, which generalize all previously known kinds. We give a complete axiomatization of algebraic dependencies in terms of simple algebraic rewriting rules. In the process we characterize exactly the expressive power of tableaux, thus solving an open problem of Aho, Sagiv and Ullman; we show that it is NP-complete to tell whether a tableau is realizable by an expression; and we give an interesting dual interpretation of the chase procedure. We also show that algebraic dependencies over a language augmented to contain union and set difference can express arbitrary domain-independent predicates of finite index over finite relations. The class of embedded implicational dependencies recently-and independently-introduced by Fagin is shown to coincide with our algebraic dependencies. Based on this, we give a simple proof of Fagin's Armstrong relation theorem.

TM-194
Barendregt, H., Longo, G.
RECURSION THEORETIC OPERATORS AND MORPHISMS ON NUMBERED SETS
Pages: 16 February 1981 $3.00
Keywords: operators, morphisms, numbered sets, recursion
Abstract: An operator is a map $\Phi:P\omega \mp P\omega$. By embedding $P\omega$ in two natural ways into the λ-calculus model $P\omega^2$ (and T^ω) the computable maps on this latter structure induce several classes of recursion theoretic operators.

TM-195
Barber, G.
RECORD OF THE WORKSHOP ON RESEARCH IN OFFICE SEMANTICS
Pages: 15 February 1981 $3.00
Keywords: office automation, knowledge-based systems
Abstract: This paper is a compendium of the ideas and issues presented at the Chatham Bars Workshop on Office Semantics. The intent of the workshop was to examine the state of the art in office systems and to elucidate the issues system designers were concerned with in developing next generation office systems. The workshop involved a cross-section of people from government, industry and academia. Presentations in the form of talks and video tapes were made of prototypical systems.

TM-196
Bhatt, S.N.
ON CONCENTRATION AND CONNECTION NETWORKS
Pages: 69 S.M. Thesis/March 1981 $3.00
Keywords· switching networks, rearrangeable concentrators, expanders, superconcentrators, probabilistic constructions, incrementally non-blocking connectors
Abstract. This thesis deals with the structural complexity of switching networks which realize concentration and connection requests when operated in a rearrangeable or incremental manner. Some of the important results and constructions are briefly reviewed. On the basis of non-constructive proof techniques used to obtain linear upper bounds on the complexity of rearrangeable concentrators, it is shown that not only are certain random graphs likely to be rearrangeably non-blocking concentrators, but that if a randomly constructed graph is not non-blocking, then, on the average, only a constant number of edges need be added to the graph to make it non-blocking. Although the problem of recognizing non-blocking networks appears to be a computationally hard problem, the extra edges may be added to the graph efficiently, during operation of the network. Finally, we obtain a constructive as well as an improved non-constructive upper bound on the complexity of incrementally non-blocking connection networks.

TM-197
Fredkin, E., Toffoli, T.
CONSERVATIVE LOGIC
Pages: 28 May 1981 $3.00
Keywords: conservative logic, reversible computing, computation universality, automata, networks, physical computing, information mechanics, discrete mechanics
Abstract: Conservative logic is a comprehensive model of computation which explicitly reflects a number of fundamental principles of physics, such as the reversibility of the dynamical laws and the conservation of certain additive quantities. Because of its closer adherence to physics than found in traditional models of computation, conservative logic is in a better position to provide indications concerning the realizations of high performance computing systems, i.e., of systems that make very efficient use of the "computing resources" actually offered by nature. In particular, conservative logic shows that it is ideally possible to build sequential circuits with zero internal power dissipation.

TM-198
Halpern, J.Y., Reif, J.H.
THE PROPOSITIONAL DYNAMIC LOGIC OF DETERMINISTIC, WELL-STRUCTURED PROGRAMS
Pages: 39 March 1981 $3.00
Keywords. propositional dynamic logic, decision procedure, polynomial space complete, expressiveness, well-structured programs
Abstract: We consider a restricted propositional dynamic logic, Strict Deterministic Propositional Dynamic Logic (SDPDL), which is appropriate for reasoning about deterministic well-structured programs. In contrast to PDL, for which the validity problem is known to be complete in deterministic exponential time, the validity problem for SDPDL is shown to be polynomial space complete. We also show that SDPDL is less expensive than PDL. The results rely on structure theorems for models of satisfiable SDPDL formulas, and the proofs give insight into the effects of nondeterminism on intractability and expressiveness in program logics.

TM-199
Mayr, E.W., Meyer, A.R.
THE COMPLEXITY OF THE WORD PROBLEMS FOR COMMUTATIVE SEMIGROUPS AND POLYNOMIAL IDEALS
Pages: 32 June 1981 $3.00
Keywords: word problem, commutative semigroup, computational complexity, polynomial ideal, exponential space, vector replacement system, Petri nets
Abstract: Any decision procedure for the word problems for commutative semigroups and polynomial ideals inherently requires computational storage space growing exponentially with the size of the problem instance to which the procedure is applied. This bound is achieved by a simple procedure for the semigroup problem.

TM-200
Burke, G.
LSB MANUAL
Pages: 92 June 1981 $3.00
Keywords: layered system building, LISP, NIL, LSB, MACLISP
Abstract: **LSB** (for Layered System Building) is an integrated set of facilities for aiding in the construction of highly-modular, multi-layered, implementation-independent LISP systems. It provides for conditional inclusion of source text, documentation production, automated declarations, and "high-level" definitions. LISP code compiled with **LSB** in general does not require **LSB** in its run-time environment. **LSB** has been in use for some time in PDP-10 Maclisp, is operational in Multics Maclisp and Lisp Machine Lisp, and is being developed for NIL.

TM-201
Meyer, A.R.
WHAT IS A MODEL OF THE LAMBDA CALCULUS? EXPANDED VERSION
Pages. 40 July 1981 $3.00
Keywords. lambda calculus, combinatory algebra, arithmetical, mathematics
Abstract: An elementary, purely algebraic definition of model for the untyped lambda calculus is given. This definition is shown to be equivalent to the natural semantic definition based on environments. These definitions of model are consistent with, and yield a completeness theorem for, the standard axioms for lambda convertibility A simple construction of models for lambda calculus is reviewed. The algebraic formulation clarifies the relation between combinators and lambda terms.

TM-202
Saltzer, J H.
COMMUNICATION RING INITIALIZATION WITHOUT CENTRAL CONTROL
Pages: 14 December 1981 $3.00
Keywords: ring initialization, communication ring, networks
Abstract: This short memorandum describes a novel combination of three well-known techniques; the combination provides a systematic way of initializing a local-area ring network without previous, static designation of a distinguished station The result is a distributed algorithm that dynamically designates a distinguished station from among a group of stations whose ability to communicate is hampered by the fact that the ring is not yet initialized. An appendix describes how this approach could be implemented as part of the 10 Megabit/second (version 2) ring network currently being installed at the M.I.T. Laboratory for Computer Science

TM-203
Bawden, A , Burke, G., Hoffman, C.W
MACLISP EXTENSIONS
Pages: 66 July 1981 $3.00
Keywords: MACLISP, LISP, NIL
Abstract: This document describes a common subset of selected facilities available in Maclisp and its derivatives: PDP-10 and Multics Maclisp, Lisp Machine Lisp (Zetalisp), and NIL. The object of this document is to aid people in writing code which can run compatibly in more than one of these environments.

TM-204
Halpern, J.Y.
ON THE EXPRESSIVE POWER OF DYNAMIC LOGIC, II
Pages: 15 August 1981 $3.00
Keywords: dynamic logic, nondeterminism, deterministic dynamic logic
Abstract: In this paper we study the expressive power of nondeterminism in dynamic logic. In particular, we show that first-order regular dynamic logic without equality (hereafter abbreviated DL) is more expressive than its deterministic counterpart (DDL). This result has already been shown for the quantifier-free case [MW], and for the propositional case [HR]. Berman and Tiuryn have recently extended the present result to the case with equality. By contrast, Meyer and Tiuryn have shown [MT] that in the r.e. case, deterministic dynamic logic coincide.

The proof hinges on showing that in a precise sense a deterministic regular program cannot search a full binary tree. Because of this, the truth of a first-order DDL formula, even with first-order quantification, cannot depend on every value in a full binary tree. From this it will follow that DDL is less expressive than DL. The kernel of the proof presented here can already be found in [HR].

TM-205
Kannon, R.
CIRCUIT-SIZE LOWER BOUNDS AND NON-REDUCIBILITY TO SPARSE SETS
Pages 22 October 1981 $3.00
Keywords: circuit size
Abstract: As remarked in Cook (1980), we do not know any nonlinear lower bound on the circuit-size of a language in P or even in NP. The best known lower bound seems to be due to Paul (1975). In this paper we show that first for each nonnegative integer k, there is a language L_k in $\Sigma_2 \psi \pi_2$ (of Meyer and Stockmeyer (1972) hierarchy) which does not have $0(n^k)$-size circuits. Using the same techniques, one is able to prove several similar results. For example, we show that for each nonnegative integer k, there is a language L_k in NP that does not have $0(n^k)$-size uniform circuits. This follows as a corollary of a stronger result shown in the paper.

Finally, we note that existence of "small circuits" is insuitable contexts equivalent to being reducible to sparse sets. Using this, we are able to prove for example that for any time-constructible super-polynomial function f(n), NTIME(f(n)) contains a language which is not many-to-one p-time reducible to any sparse set.

TM-206
Leiserson, C.E., Pinter, R.Y.
OPTIMAL PLACEMENT FOR RIVER ROUTING
Pages: 16 October 1981 $3 00
Keywords: algorithms, graph theory, placement and routing, river routing, VLSI
Abstract: Programs for integrated circuit layout typically have two phases: placement and routing. The router should produce as efficient a layout as possible, but of course the quality of the routing depends heavily on the quality of the placement. On the other hand, the placement procedure ideally should know the quality of a routing before it routes the wires. In this talk we present an optimal solution for a practical, common version of this placement and routing problem.

River routing is the problem of connecting in order a set of terminals $a_1,...,a_n$ on a line to another set $b_1,...,b_n$ across a rectangular channel. Since the terminals are located on modules, the modules must be placed relative to one another before routing. This placement problem arises frequently in design systems like bristle-blocks where stretch lines through a module can effectively break into several chunks, each of which must be placed separately. In this talk we shall present concise necessary and sufficient conditions for wirability which are applied to reduce the optimal placement problem to the graph-theoretic single-source-longest-paths problem. By exploiting the special structure of graphs that arise from the placement problem for rectilinear wiring, an optimal solution may be determined in linear time.

TM-207
Longo, G.
POWER SET MODELS OF LAMBDA-CALCULUS: THEORIES, EXPANSIONS, ISOMORPHISMS
Pages: 36 November 1981 $3.00
Keywords: mathematics, lambda calculus, power set models
Abstract: This paper mainly deals with the models for type free λ-calculus defined by Plotkin and Engeler. Section 1 (part 1) introduces λ-terms and CL-terms (terms of λ-calculus, $\lambda\beta$, and of Combinatory Logic, CL) of various orders corresponding to levels of functionality or number of λ-abstractions. Part 2 discusses the consequences in Combinatory Algebras of an early remark of Wadsworth (and Scott) on how to interpret the "loss of information" which is implicit in performing combinatory reductions, as in any effective process.

Section 2, following Barendregt's terminology, deals with the local analysis of Engeler's models, i.e syntactically characterizes the true equalities in these models. This provides an algebraic characterization of λ-terms possessing normal form.

Section 3 gives a semantical characterization of f4l-terms of any finite (and infinite) order.

Section 4 contains the model-theoretic applications of this paper

TM-208
Cosmadakis, S.S., Papadimitriou, C.H.
THE TRAVELING SALESMAN PROBLEM WITH MANY VISITS TO FEW CITIES
Pages: 21 November 1981 $3.00
Keywords: traveling salesman problem, dynamic programming, assignment problem, minimal Eulerian digraph, Stirling's formula, min-cost max-flow problem, Edmonds-Karp scaling method
Abstract: We study the version of the traveling salesman problem in which a relatively small number of cities —say, six— must be visited a huge number of times —e.g., several hundred times each. (It costs to go from one city to itself.) We develop an algorithm for this problem whose running time is exponential in the number of cities, but logarithmic in the number of visits. Our algorithm is a practical approach to the problem for instances of size in the range indicated above. The implementation and analysis of our algorithm give rise to a number of interesting graph-theoretic and counting problems.

TM-209
Johnson, D., Papadimitriou, C.H.
COMPUTATIONAL COMPLEXITY AND THE TRAVELING SALESMAN PROBLEM
Pages: 61 December 1981 $3.00
Keywords: traveling salesman problem, computational complexity, polynomial-time, P, NP, and NP-complete problems, Hamilton circuit
Abstract: In the last decade or so a theory of computational complexity has developed, based on rigorous methods for evaluating algorithms and for classifying problems as "hard" or "easy". This theory is deeply indebted to the field of Combinatorial Optimization, which has provided it with invaluable motivation, insight, and paradigms. The TSP is probably the most important among the latter. It has served as a testbed for about every new algorithmic idea, and was one of the first optimization problems conjectured to be "hard" in a specific technical sense.

In this chapter we shall be studying the complexity of the TSP, while providing an introduction to the general area of computational complexity. Section 1 provides an introduction to the notion of an algorithm and to different measures of its "complexity". Section 2 shows how we can restrict our attention to decision problems (problems with a "yes" or "no" answer), and divide these into equivalence classes according to their difficulty. Section 3 then proves that even very restricted versions of the TSP are likely to be very hard (they are, in the technical jargon, NP-complete). Section 4 concludes the chapter by examining the complexity of problems related to the TSP, as well as complexity aspects of various proposed approaches to the TSP.

TM-210
Greif, I.
SOFTWARE FOR THE 'ROLES' PEOPLE PLAY
Pages: 13 February 1983 $3.00
Keywords: office automation, desk-to-desk conferencing
Abstract: Office work consists largely of cooperative efforts by numbers of people. To support such work, applications programs can be designed as "multi-person" systems organized around notions of "roles" and "working relationships". A group of co-workers can then describe to the system their agreed upon roles in a project as well as the working relationships among these roles. Based on this description, application software can provide support for communications protocols and access control that is tailored to the working situation. As working relationships evolve, there descriptions can be modified so that the software will continue to meet the needs of the users.

The paper presents an approach to office systems research emphasizing the development of software modules that can be used to build end-user application programs. The requirements that "multi-person" applications place on this software architecture are discussed in the context of a series of examples of multi-person activities, including joint document writing and calendar management.

TM-211
Meyer, A.R., Tiuryn, J.
A NOTE ON EQUIVALENCES AMONG LOGICS OF PROGRAMS
Pages: 17 December 1981 $3.00
Keywords: logics of programs, dynamic logic, logic of effective definitions
Abstract: Several different first order formal logics of programs — Algorithmic Logic, Dynamic Logic, and Logic of Effective Definitions — are compared and shown to be equivalent to a fragment of constructive $L_{\omega 1 \omega}$. When programs are modeled as effective flowcharts, the logics of deterministic and nondeterministic programs are equivalent.

TM-212
Elias, P.
MINIMAX OPTIMAL UNIVERSAL CODEWORD SETS
Pages: 23 January 1982 $3.00
Keywords: codeword sets, encoding
Abstract: In an interactive multi-user data-processing system a user knows the probabilities of his messages and must encode them into a fixed system-wide variable-length codeword set. He needs to receive the answer to his last message before selecting the next, so his encoding is one shot. To minimize average codeword length he encodes his messages in order of decreasing probability into codewords in order of increasing length. I give an algorithm which, for each of several measures of performance, finds the codeword set best by that measure for the worst user, and some of the minimax optimal codeword sets the algorithm has found. Some of the results hold for all user distributions: others require e.g. that all users send exactly or at most *m* distinct messages, and/or that there is an integer *k* such that no user has a message of probability greater than 1/*k*.

TM-213
Greif, I.
PCAL: A PERSONAL CALENDAR
Pages: 29 January 1982 $3.00
Keywords: office automation, calendar management, personal data base
Abstract: This paper is a users guide to PCAL, the personal calendar program available on the MIT-LCS DEC-20/60. The program can be used to create and maintain a personal data base of calendar information including appointments, meetings, classes (and other regular occurring meetings) and notes.

TM-214
Meyer, A.R., Mitchell, J.C.
TERMINATIONS FOR RECURSIVE PROGRAMS: COMPLETENESS AND AXIOMATIC DEFINABILITY
Pages: 33 March 1982 $3.00
Keywords: axiomatic definitions of programming languages, complete axiomatization, semantics of programming languages, termination assertions, recursive procedures
Abstract: The termination assertion $p<\mathbf{S}>q$ means that whenever the formula p is true, there is an execution of the poly nondeterministic program **S** which terminates in a state in which q is true. A recursive program **S** may declare and use local variables and nondeterministic recursive procedures with call-by-address and call-by-value parameters, in addition to accessing undeclared variables and global procedures. Assertions p and q about calls to global procedures are first order formulas extended to express hypotheses about the termination of calls to undeclared global procedures. A complete, effective axiom system with axioms corresponding to the syntax of the programming language isgiven for the termination assertions valid over all interpretations. Termination assertions define the semantics of recursive programs in the following sense: if two programs have different input-output semantics, then there is a termination assertion that is valid for one program but not for the other. Thus the complete axiomatization of termination assertions constitutes an axiomatic definition of the semantics of recursive programs.

TM-215
Leiserson, C.E., Saxe, J.B.
OPTIMIZING SYNCHRONOUS SYSTEMS
Pages: 24 March 1982 $3.00
Keywords: circuit optimization, design methodology, graph theory, parallel computation, path algorithms, pipelining, synchronous systems, systolic systems, VLSI
Abstract: The complexity of integrated-circuit chips produced today makes it feasible to build inexpensive, special-purpose subsystems that rapidly solve sophisticated problems on behalf of a general-purpose host computer. This paper contributes to the design methodology of efficient VLSI algorithms. We present a transformation that converts synchronous systems into more time-efficient, *systolic* implementations by removing combinational rippling.

The problem of determining the optimized system can be reduced to the graph-theoretic *single-destination-shortest-paths* problem More importantly from an engineering standpoint, however, the kinds of rippling that can be removed from a circuit at essentially no cost can be easily characterized. For example, if the global communication in a system is *broadcastingO from the host computer, the broadcast can always be replaced by local communication.*

TM-216
Church, K.W., Patil, R.S.
COPING WITH SYNTACTIC AMBIGUITY OR HOW TO PUT THE BLOCK IN THE BOX ON THE TABLE
Pages: 36 April 1982 $3.00
Keywords: natural language, parsing, ambiguity
Abstract: Sentences are far more ambiguous than one might have thought. There may be hundreds, perhaps thousands of syntactic parse trees for certain very natural sentences of English. This fact has been a major problem confronting natural language processing because it indicates that it may require a long time to construct a list of all the parse trees, and furthermore, it isn't clear what to do with the list once it has been constructed. This list may be so numerous that it is probably not the most convenient representation for communication with the semantic and pragmatic processing modules. In this paper we propose some methods for dealing with syntactic ambiguity in ways that take advantage of certain regularities among the alternative parse trees. These regularities will be expressed as **linear combinations** of ATN networks, and also as sums and products of formal power series.

We suggest some ways that practical processor can take advantage of this modularity in order to deal more efficiently with combinatoric ambiguity. In particular, we will show how a processor can efficiently compute the ambiguity of an input sentence (or any portion thereof). Furthermore, we will show how to compile certain grammars into a form that can be processed more efficiently. In some cases, including the "every way ambiguous" grammar (e.g. conjunction, prepositional phrases, noun-noun modification), processing time will be reduced from $0(n^3)$ to 0(n). Finally, we will show how to uncompile certain highly optimized grammars into a form suitable for linguistic analysis.

TM-217
Wright, K.D.
A FILE TRANSFER PROGRAM FOR A PERSONAL COMPUTER
Pages: 42 April 1982 $3.00
Keywords: file transfer, personal computers, concurrency
Abstract This thesis explores engineering decisions involved in implementing a network file transfer program on a personal computer in response to criteria of low cost and reasonable efficiency. The issues include choice of hardware, design of the network, choice of implementation language, choice of communication protocols, and choice of structure A machine level protocol is designed. A project incorporating these and other ideas is undertaken and the ideas thus evaluated. Insight is gleaned into the performance expected under varying operating system and interrupt environments. A notion of an "ideal" operating system interface for applications similar to file transfer (which can exploit concurrency) is developed. Finally, possible improvements on the actual project are suggested based in part on efficiency data obtained.

TM-218
Greif, I.
COOPERATIVE OFFICE WORK, TELECONFERENCING AND CALENDAR MANAGEMENT: A COLLECTION OF PAPERS
Pages: 44 May 1982 $3 00
Keywords: office automation, calendar management, desk-to-desk conferencing, teleconferencing
Abstract: This technical memo consists of a collection of papers that have been presented at conferences. They all present results of research in the "Multi-person Informational Work" project in the Office Automation Group.

TM-219
Jouannaud, J.-P., Lescanne, P., Reinig, F.
RECURSIVE DECOMPOSITION ORDERING AND MULTISET ORDERINGS
Pages· 29 June 1982 $3.00
Keywords: termination, term rewriting systems, equational theories, well-founded ordering, simplification, multiset ordering
Abstract: The Recursive Decomposition Ordering, a simplification ordering on terms, is useful to prove termination of term rewriting systems. In this paper we give the definition of the decomposition ordering and prove that it is a well-founded simplification ordering containing Dershowitz's Recursive Path Ordering. We also show that the Recursive Decomposition Ordering has a very interesting incremental property. In the second paper, we propose two well-founded orderings on multisets that extend the Dershowitz-Manna ordering. Unlike the Dershowitz-Manna ordering, ours do not have a natural monotonicity property. This lack of monotonicity suggests using monotonicity to provide a new characterization of the Dershowitz-Manna ordering. Section 5 proposes an efficient and correct implementation of that ordering.

TM-220
Chu, T.-A.
CIRCUIT ANALYSIS OF SELF-TIMED ELEMENTS FOR NMOS VLSI SYSTEMS
Pages: 40 May 1982 $3.00
Keywords: self-timed systems, dual-rail code, resetting signaling, C-element, meta-stable
Abstract: Scaling of VLSI digital systems introduces new problems to the design of synchronous systems, due to the disproportional increase in wire delays with the decrease in transistor sizes. On the other hand, the asynchronous self-timed design approach, which has been traditionally less attractive, offer a number of advantages for VLSI. Also, this approach can be directly incorporated into a structured design methodology for Packet Communication Architectures. This paper considers a practical self-timed design methodology and studies its implementation in nMOS. The C-element and the arbiter circuit components of self-timed systems, are analyzed to allow the evaluation of the design approach.

TM-221
Leighton, F.T., Lepley, M., Miller, G.L.
LAYOUTS FOR THE SHUFFLE-EXCHANGE GRAPH BASED ON THE COMPLEX PLANE DIAGRAM
Pages: 21 June 1982 $3.00
Keywords: area-efficient chip layouts, complex plane diagram, graph embedding, shuffle-exchange graph, Thompson grid model, VLSI
Abstract: The shuffle-exchange graph is one of the best structures known for parallel computation. Among other things, a shuffle-exchange computer can be used to compute discrete Fourier transforms, multiply matrices, evaluate polynomials, perform permutations and sort lists. The algorithms needed for these operations are extremely simple and many require no more than logarithmic time and constant space per processor. In this paper, we analyze the algebraic structure of the shuffle exchange graph in order to find area-efficient embeddings of the graph in a two-dimensional grid. The results are applicable to the design of VLSI circuit layouts for a shuffle-exchange computer.

TM-222
Meier zu Sieker, F.
A TELEX GATEWAY FOR THE INTERNET
Pages: 59 S.B. Thesis/May 1982 $3.00
Keywords: local computer networks, networks, Telex
Abstract: The design of a gateway connecting one of the networks of the MIT Laboratory for Computer Science to the telex network is discussed. A description of the telex network is given. The relationship of the gateway to other resources of the network environment is considered to obtain directions for the implementation of new resources. The implementation of the gateway to the UNIX operating system outlined.

TM-223
diSessa, A.A.
A PRINCIPLED DESIGN FOR AN INTEGRATED COMPUTATIONAL ENVIRONMENT
Pages: 43 July 1982 $3.00
Keywords: spatial metaphor, user models, understandability, programming languages, text editor, data base language, integrated computational environment, novice programming
Abstract: Boxer is a computer language designed to be the base of an integrated computational environment providing a broad array of functionality — from text editing — for naive and novice users. It stands in the line of LISP inspired languages (LISP, LOGO, SCHEME), but differs from these in achieving much of its understandability from pervasive use of a spatial metaphor reinforced through suitable graphics. This paper describes a set of learnability and understandability issues first and then uses them to motivate design decisions made concerning Boxer and the environment in which it is embedded.

TM-224
Barber, G.
SUPPORTING ORGANIZATIONAL PROBLEM SOLVING WITH A WORKSTATION
Pages: 29 July 1982 $3.00
Keywords: problem solving, workstations, omega, viewpoints, office semantics, change and contradiction, office automation
Abstract: This paper describes an approach to supporting work in the office. Using and extending ideas from the field of AI we describe office work as a problem solving activity. A knowledge embedding language called OMEGA is used to embed knowledge of the organization into an office worker's workstation in order to support the office worker in his or her problem solving. A particular approach to reasoning about change and contradiction is discussed. This approach uses Omega's viewpoint mechanism.

Omega's viewpoint mechanism is a general contradiction handling facility. Unlike other Knowledge Representation systems, when a contradiction is reached the reasons for the contradiction can be analyzed by the deduction mechanism without having to resort to a backtracking mechanism.

The Viewpoint mechanism is the heart of the Problem Solving Support Paradigm. This paradigm supplements the classical AI view of problem solving. Office workers are supported using the Problem Solving Support Paradigm.

An example is presented where Omega's facilities are used to support an office worker's problem solving activities. The example illustrates the use of viewpoints and of Omega's capabilities to reason about it's own reasoning processes.

TM-225
Barber, G., Hewitt, C.
FOUNDATIONS FOR OFFICE SEMANTICS
Pages: 19 July 1982 $3.00
Keywords: office semantics, problem solving, office automation
Abstract: In this paper we develop the semantics of work in the office in terms of the concepts of application structure and organizational structure of the office. Application structure is concerned with the rules and constraints of the domain of the office work such as accounting, law, or social relationships within the organization. Detailed knowledge of office application structures and organizational structures is necessary in order to understand how they interact and evolve.

Problem solving is a pervasive activity within offices which is performed when office workers apply general knowledge about office procedures to the specific cases encountered in their daily work.

We discuss how a description system (named OMEGA) can aid in the construction of interactive systems whose intent is to describe the application and organization structures. Using the knowledge embedded within itself about the office OMEGA can help support office workers in their problem solving processes.

TM-226
Bergstra, J., Chmielinska, A., Tiuryn, J
HOARES'S LOGIC IS NOT COMPLETE WHEN IT COULD BE
Pages 14 August 1982 $3.00
Keywords. Hoare Logic, partial correctness, relative completeness
Abstract. It is known (cf.[2]) that if the Hoare rules are complete for a first-order structure A, then the set of partial correctness assertions true over A is recursive in the first-order theory of A. We show that the converse is not true. Namely, there is a first-order structure Ξ such that the set of partial correctness assertions true over Ξ is recursive in the theory of Ξ, but Hoare rules are not complete for Ξ.

TM-227
Leighton, F.T.
NEW LOWER BOUND TECHNIQUES FOR VLSI
Pages· 30 August 1982 $3 00
Keywords: area-efficient chip layouts, bisection, graph embedding, parallel computation, Thompson grid model, tree of meshes, VLSI, wire length
Abstract. In this paper, we use crossing number and wire area arguments to find lower bounds on the layout area and maximum edge length of a variety of new computationally useful networks. In particular, we describe

1) an N-node planar graph which has layout area $\Theta(\mathrm{N}/log\mathrm{N})$ and maximum edge length $\Theta z(\mathrm{N}^{1/2}/log^{1/2}\mathrm{N})$,

2) an N-node graph with an $0(x^{1/2})$-separator which has layout area $\Theta(\mathrm{N}/log^2\mathrm{N})$ and maximum edge length $\Theta(\mathrm{N}^{1/2}log\mathrm{N}/loglog\mathrm{N})$, and

3) an N-node graph with an $0(x^{1-1/r})$-separator which has maximum edge length $\Theta z(\mathrm{N}^{1-1/r})$ for any $r \geq 3$.

TM-228
Papadimitriou, C H , Zachos, S.K.
TWO REMARKS ON THE POWER OF COUNTING
Pages: 10 August 1982 $3.00
Keywords: counting problems, oracle computation, polynomial hierarchy, parity problems, machine simulation
Abstract· The relationship between hierarchy and Valiant's class #P is at present unknown. We show that some low portions of the polynomial hierarchy, namely deterministic polynomial algorithms using an NP oracle at most a logarithmic number of times, can be simulated by one #P computation We also show that the class of problems solvable by polynomial-time nondeterministic Turing machines which accept whenever there is an *odd* number of accepting computations is idempotent, that is, closed under usage of oracles from the same class.

TM-229
Cosmadakis, S.S
THE COMPLEXITY OF EVALUATION RELATIONAL QUERIES
Pages 14 August 1982 $3.00
Keywords· relational data base, relational expression, completeness, hardness, enumeration problem, polynomial-time, NP and co-NP
Abstract· We show that, given a relation R, a relational query φ involving only projection and join, and a conjectured result r, testing whether $\varphi(\mathrm{R}) = r$ is D^p-complete. Bounding the size of $\varphi(\mathrm{R})$ from below (above) is *NP*-hard (co-NP-hard), and bounding it both ways is D^p-hard. Computing the size of $\varphi(\mathrm{R})$ is #*P*-hard.

We also show that, given two relations R_1 and R_2 and two queries φ_1 and φ_2 as above, testing whether $\varphi_1(\mathrm{R}_1) \quad \varphi_2(\mathrm{R}_2)$ and testing whether $\varphi_1(\mathrm{R}_1) = \varphi_2(\mathrm{R}_2)$ are both $\pi_2{}^{\mathrm{p}}$-complete, even when $\mathrm{R}_1 = \mathrm{R}_2$ or when $\varphi_1 = \varphi_2$.

TM-230
Shamir, A.
EMBEDDING CRYPTOGRAPHIC TRAPDOORS IN ARBITRARY KNAPSACK SYSTEMS
Pages: 8 September 1982 $3.00
Keywords: cryptography, Merkle-Hellman cryptosystems, knapsack problems
Abstract: In this paper we show that after sufficiently many modular multiplications, any knapsack system becomes a trapdoor system that can be used in public-key cryptography.

TM-231
Kleitman, D., Leighton, F.T., Lepley, M., Miller, G.L.
AN ASYMPTOTICALLY OPTIMAL LAYOUT FOR THE SHUFFLE-EXCHANGE GRAPH
Pages: 28 October 1982 $3.00
Keywords: area-efficient chip layouts, cube-connected-cycles graph, graph embedding, shuffle-exchange graph, Thompson grid model, VLSI
Abstract: The shuffle-exchange graph is one of the best structures known for parallel computation. Among other things, a shuffle-exchange computer can be used to compute discrete Fourier transforms, multiply matrices, evaluate polynomials, perform permutations and sort lists. The algorithms needed for these operations are quite simple and many require no more than logarithmic time and constant space per processor. In this paper, we describe an $O(N^2/log^2N)$-area layout for the shuffle-exchange graph on a two-dimensional grid. The layout is the first which is known to achieve Thompson's asymptotic lower bound.

TM-232
Yeh, A.
PLY: A SYSTEM OF PLAUSIBILITY INFERENCE WITH A PROBABILISTIC BASIS
Pages: 25 December 1982 $3.00
Keywords: expert systems, plausibility inference, probabilistic inference, planning aid
Abstract: An overview is given of a system of plausibility inference that will be developed for use in planning. This system, to be called **PLY**, will be specifically designed to work with propositions of the form "when A is true (occurs), B is likely to be true (to occur)". Previous systems performing similar functions have been designed as aids for such tasks as medical diagnosis (MYCIN and others) and mineral prospecting (PROSPECTOR).

PLY will have a probabilistic basis. Intuitive assumptions to deal with knowledge not explicitly given to the system will be made with the aid of an information-theoretic measure on the amount of information in a probability distribution. Unlike many other systems, PLY will not use these assumptions when the given knowledge indicates they are not tenable. In addition to standard probabilities, PLY will be able to make use of knowledge (information) in the form of correlations and increased/decreased likelihoods, which most people find easier to estimate than probabilities.

PLY's knowledge will be in an organized and structured form, which will help in knowledge acquisition and revision, facilitate system explanations, and lower the storage requirements of the system.

TM-233
Konopelski, L.J.
IMPLEMENTING INTERNET REMOTE LOGIN ON A PERSONAL COMPUTER
Pages: 79 S.B. Thesis/December 1982 $3.00
Keywords: internet remote login, personal computers, asynchronous systems, tasking
Abstract: This thesis demonstrates that a desktop personal computer can support an efficient internet remote login implementation with the same protocols used by large mainframes. It describes a project in which the Telnet remote login protocol, along with the supporting Transmission Control Protocol and Internet Protocol were implemented on an IBM Personal Computer. The utility of the implementation depended heavily on the software speed. Strategies discussed to insure quick performance included tailoring protocols to their clients needs, sharing the overhead of asynchronous actions, and sharing data. A natural order in which to process the protocol data was identified, and two control structures were

presented that allowed the protocol modules to run in this order. One of the control structures used procedures and processes, while the other used procedures alone.

A full scale protocol was successfully placed in the personal computer. With some foreign hosts, the implementation echoed characters in less than a quarter of a second, and processed a screenful of data in less than three seconds. The protocol software overhead was never the dominating performance bottleneck. The serial line interface limited the character echoing performance while the speed with which the processor could operate its display limited the processing speed of large amounts of data. Memory size was not a significant constraint.

TM-234
Rivest, R., Sherman, A.T.
RANDOMIZED ENCRYPTION TECHNIQUES
Pages: 20 January 1983 $3.00
Keywords: cryptography, cryptology, modes of operation, randomized encryption
Abstract: A *randomized encryption procedure* enciphers a message by randomly choosing a ciphertest from a set of ciphertexts corresponding to the message under the current encryption key. At the cost of increasing the required bandwidth, such procedures may achieve greater cryptographic security than their deterministic counterparts by increasing the apparent size of the message space, eliminating the threat of chosen plaintext attacks, and improving the *a priori* statistics for the inputs to the encryption algorithms. In this paper we explore various ways of using randomization in encryption.

TM-235
Mitchell, J.C.
THE IMPLICATION PROBLEM FOR FUNCTIONAL AND INCLUSION DEPENDENCIES
Pages: 23 February 1983 $3.00
Keywords: relational data base, inclusion dependency, functional dependency, complete axiomatization, undecidability, finite implication
Abstract: There are two implication problems for functional dependencies and inclusion dependencies: general implication and finite implication. Given a set of dependencies $\Sigma \cup <\sigma>$, the problems are to determine whether σ holds in all data bases satisfying Σ or all finite data bases satisfying Σ. Contrary to the possibility suggested in [5], there is a natural, complete axiom system for general implication. However, a simple observation shows that both implication problems are recursively unsolvable. It follows that there is no recursively enumerable set of axioms for finite implication.

TM-236
Leighton, F.T., Leiserson, C.E.
WAFER-SCALE INTEGRATION OF SYSTOLIC ARRAYS
Pages: 29 February 1983 $3.00
Keywords: channel width, fault-tolerant systems, probabilistic analysis,spanning tree, systolic arrays, traveling salesman problem, tree of meshes, VLSI, wafer-scale integration, wire length
Abstract: VLSI technologists are fast developing *wafer-scale integration*. Rather than partitioning a silicon wafer into chips as is usually done, the idea behind wafer-scale integration is to assemble an entire system (or network of chips) on a single wafer, thus avoiding costs and performance loss associated with individual packing of chips. A major problem with assembling a large system of microprocessors on a single wager, however, is that some of the processors, or *cells*, on the wafer are likely to be defective. In the paper, we describe practical procedures for integrating wafer-scale systems "around" such faults. The procedures are designed to minimize the length of the longest wire in the system, thus minimizing the communication time between cells. Although the underlying network problems are NP-complete, we prove that the procedures are reliable by assuming a probabilistic model of cell failure. We also discuss applications of this work to problems in VLSI layout theory, fault-tolerant systems and planar geometry.

TM-237
Dolev, D., Leighton, F.T., Trickey, H.
PLANAR EMBEDDING OF PLANAR GRAPHS
Pages: 8 February 1983 $3.00
Keywords: graph embedding, planar graphs, outerplanar graphs, crossover-free, VLSI, NP-complete
Abstract: Planar embedding with minimal area of graphs on an integer grid is an interesting problem in VLSI theory. Valiant gave an algorithm to construct a planar embedding for trees in linear area; he also proved that there are planar graphs that requuire quadratic area. .br We fill in a spectrum between Valiant's results by showing that an *N*-node planar graph has a planar embedding with area *0(NF)*, where *F* is a bound on the path length from any node to the exterior face. In particular, an outerplanar graph can be embedded without crossovers in linear area. This bound is tight, up to constant factors: for any *N* and *F*, there exist graphs requiring Ω*(NF)* area for planar embedding.

Also, finding a minimal embedding area is shown to be *NP*-complete for forests, and hence for more general types of graphs.

TM-238
Baker, B.S., Bhatt, S.N., Leighton, F.T.
AN APPROXIMATION ALGORITHM FOR MANHATTAN ROUTING
Pages: 16 February 1983 $3.00
Keywords: algorithms, channel routing, Manhattan routing, VLSI, multipoint net
Abstract: *Density* has long been known to be an important measure of difficulty for Manhattan routing. In this paper, we identify a second important measure of difficulty, which we call *flux*. We show that flux, like density, is a lower bound on channel width. In addition, we present a linear-time algorithm which routes any multipoint net Manhattan routing problem with density *d* and flux *f* in a channel of width *2d + 0(F)*. (for 2-point nets, the bound is *d + 0(f)*.) Thus we show that Manhattan routing is one of the NP-complete problems for which there is a provably good approximation algorithm.

Since *f*≤*(sq.rt. of n)* for any *n*-net problem, the preceding bound indicates that every *n*-net problem can be routed in a channel of width 0(*d* + *(sq.rt. of n*), thus proving a conjecture of Brown and Rivest. For practical problems, however, the flux appears to be bounded by a small constant. In this case, the algorithm uses *2d* + O(1) tracks. (For 2-point nets, the bound is *d + 0(1)*.) These bounds are (asymptotically) nearly twice as good as those for the best known knock-knee algorithm and nearly as good as those for the best known 3-layer algorithm, yet do not require the use of either knock-knees or 3-layers of interconnect.

The results also have applications to a model of channel routing, which we call the *3-parameter model*, that is closer to the design rules of current fabrication technologies. The 3-parameter model is similar to the Manhattan model except that wires are assumed to be narrower than contact cuts in the 3-parameter model (as is the case in most fabrication technologies). By modifying the Manhattan routing algorithm, we show that *every* 3-parameter problem can be routed in a channel of width 2*d* + *0(1)*. (For 2-point nets, the bound is *d + 0(1)*.) Thus the 3-parameter model (like the knock-knees model) is a simple variation of Manhattan routing for which every problem can be routed in *0(d)* tracks.

TM-239
Sutherland, J.B., Sirbu, M.
EVALUATION OF AN OFFICE ANALYSIS METHODOLOGY
Pages: 17 March 1983 $3.00
Keywords: office automation, automated office systems, office analysis, understanding office work, office model, methodology evaluation
Abstract: We have developed a model of the office that describes semi-structured office work. This model underlies an office analysis methodology and an office specification language. An evaluation of the usefulness and practicality of the model, the specification language, and the methodology has shown that the model is clearly a useful approach to understanding offices, the specification language is interesting but not as useful in practice as we had hoped, and the methodology is useful but could be improved. We have developed a new methodology that addresses the issue of diagnosis as well as description. This new methodology is still being evaluated, but early results show that it is as useful for training new analysts as the old methodology.

TM-240
Bromley, Hank
A PROGRAM FOR THERAPY OF ACID-BASE AND ELECTROLYTE DISORDERS
Pages: 35 S.B. Thesis/June 1983 $3.00
Keywords: medical applications, clinical decision making, expert systems
Abstract: This thesis describes work done on the therapy component of an on-going project for the diagnosis and management of acid-base and electrolyte disorders. Therapeutic interventions can be classified as *symptomatic* or *etiologic*, and as *acute* or *chronic*. We have focused on the problem of acute symptomatic therapy. Based on observation of clinical

practice, we have developed a formalization of the domain-independent aspect of the task of acute symptotic therapy, then applied the formalization to the particular field of acid-base and electrolyte disorders. A rule-based program named ABET (the **A**cid-**B**ased and **E**lectrolyte **T**herapy Advisor) has been designed and written to test this formalization.

The thesis presents the methods used by ABET, the program's implementation, a sample session, and a discussion of limitations and possible improvements.

AUTHOR INDEX

AUTHOR INDEX

Abelson, H.	LOWER BOUNDS ON INFORMATION TRANSFER IN DISTRIBUTED COMPUTATIONS	TM-102
Ackerman, W.B.	RESEARCH DIRECTIONS IN COMPUTER ARCHITECTURE	TM-114
Ackerman, W.B.	VAL--A VALUE-ORIENTED ALGORITHMIC LANGUAGE, PRELIMINARY REFERENCE MANUAL	TR-218
Ackerman, W.B.	A STRUCTURE MEMORY FOR DATA FLOW COMPUTERS	TR-186
Adelman, L.M.	SPACE-BOUNDED SIMULATION OF MULTITAPE TURING MACHINES	TM-148
Adelman, L.M.	TIME, SPACE AND RANDOMNESS	TM-131
Adelman, L.M.	MENTAL POKER	TM-125
Adelman, L.M.	A METHOD FOR OBTAINING SIGNATURES AND PUBLIC-KEY CRYPTOSYSTEMS	TM-082
Aiello, J.M.	AN INVESTIGATION OF CURRENT LANGUAGE SUPPORT FOR THE DATA REQUIREMENTS OF ...	TM-051
Alsop, J.W.	A CANONIC TRANSLATOR	TR-046
Alter, S.	A VERY HIGH LEVEL LANGUAGE FOR BUSINESS DATA PROCESSING	TR-254
Altman, V.E.	A LANGUAGE IMPLEMENTATION SYSTEM	TR-126
Amikura, K.	A LOGIC DESIGN FOR THE CELL BLOCK OF A DATA FLOW PROCESSOR	TM-093
Anderson, T.L.	THE DESIGN OF A MULTIPROCESSOR DEVELOPMENT SYSTEM	TR-279
Aoki, D.J.	A MACHINE LANGUAGE INSTRUCTION SET FOR A DATA FLOW PROCESSOR	TM-146
Archer, R.F.	REPRESENTATION AND ANALYSIS OF REAL-TIME CONTROL STRUCTURES	TR-241
Arens, G.C.	RECOVERY OF THE SWALLOW REPOSITORY	TR-252
Arvind	I-STRUCTURES: AN EFFICIENT DATA TYPE FOR FUNCTIONAL LANGUAGES	TM-178
Arvind	A DATA FLOW ARCHITECTURE WITH TAGGED TOKENS	TM-174
Atkinson, R.R.	AUTOMATIC VERIFICATION OF SERIALIZES	TR-229
Atkinson, R.R.	CLU REFERENCE MANUAL	TR-225
Baecker, R.M.	INTERACTIVE COMPUTER-MEDIATED ANIMATION	TR-061
Bagchi, A.	ECONOMY OF DESCRIPTIONS AND MINIMAL INDICES	TM-027
Baker, B.S.	AN APPROXIMATION ALGORITHM FOR MANHATTAN ROUTING	TM-238
Baker, C.M.	ARTWORK ANALYSIS TOOLS FOR VLSI CIRCUITS	TR-239
Baker, H.G.	ACTOR SYSTEMS FOR REAL-TIME COMPUTATION	TR-197
Baker, H.G.	ACTORS AND CONTINUOUS FUNCTIONALS	TR-194
Banks, E.R.	INFORMATION PROCESSING AND TRANSMISSION IN CELLULAR AUTOMATA	TR-081
Baratz, A.E.	AN ANALYSIS OF THE SOLOVAY AND STRASSEN TEST FOR PRIMALITY	TM-108
Baratz, A.E.	CONSTRUCTION AND ANALYSIS OF NETWORK FLOW PROBLEM WHICH FORCES KARZANOV ...	TM-083
Baratz, A.E.	THE COMPLEXITY OF THE MAXIMUM NETWORK FLOW PROBLEM	TR-230
Barber, G.	FOUNDATIONS FOR OFFICE SEMANTICS	TM-225
Barber, G.	SUPPORTING ORGANIZATIONAL PROBLEM SOLVING WITH A WORKSTATION	TM-224
Barber, G.	RECORD OF THE WORKSHOP ON RESEARCH IN OFFICE SEMANTICS	TM-195
Barendregt, H.	RECURSION THEORETIC OPERATORS AND MORPHISMS ON NUMBERED SETS	TM-194
Barnett, A.I.	A COMPUTER SYSTEM FOR DECISION ANALYSIS IN HODGKINS DISEASE	TR-271
Bawden, A.	MACLISP EXTENSIONS	TM-203
Bayer, P.J.	IMPROVED BOUNDS ON THE COSTS OF OPTIMAL AND BALANCED BINARY SEARCH TREES	TM-069
Ben-Ari, M.	DETERMINISTIC PROPOSITIONAL DYNAMIC LOGIC: FINITE MODELS, COMPLEXITY, AND ...	TM-190
Benedict, G.G.	AN ENCIPHERING MODULE FOR MULTICS	TM-050
Bengelloun, S.A.	MDC-PROGRAMMER: A MUDDLE-TO-DATALANGUAGE TRANSLATOR FOR INFORMATION ...	TM-053
Benjamin, A.J.	IMPROVING INFORMATION STORAGE RELIABILITY USING A DATA NETWORK	TM-078
Berez, J.M.	A DYNAMIC DEBUGGING SYSTEM FOR MDL	TM-094
Bergstra, J.	HOARES'S LOGIC IS NOT COMPLETE WHEN IT COULD BE	TM-226
Berzins, V.A.	ABSTRACT MODEL SPECIFICATIONS FOR DATA ABSTRACTIONS	TR-221
Beyer, W.T.	RECOGNITION OF TOPOLOGICAL INVARIANTS BY ITERACTIVE ARRAYS	TR-066
Bhatt, S.N.	AN APPROXIMATION ALGORITHM FOR MANHATTAN ROUTING	TM-238
Bhatt, S.N.	ON CONCENTRATION AND CONNECTION NETWORKS	TM-196
Biggs, J.M.	STRESS: A PROBLEM-ORIENTED LANGUAGE FOR STRUCTURAL ENGINEERING	TR-006
Bishop, P.B.	COMPUTER SYSTEMS WITH A VERY LARGE ADDRESS SPACE AND GARBAGE COLLECTION	TR-178
Blank, M.	THE MDL PROGRAMMING LANGUAGE PRIMER	TR-292
Bloniarz, P.A.	A NOTE ON THE AVERAGE TIME TO COMPUTE TRANSITIVE CLOSURES	TM-076
Bloniarz, P.A.	THE COMPLEXITY OF MONOTONE BOOLEAN FUNCTIONS AND AN ALGORITHM FOR FINDING ...	TR-238
Bloom, T.	CLU REFERENCE MANUAL	TR-225
Bloom, T.	SYNCHRONIZATION MECHANISMS FOR MODULAR PROGRAMMING LANGUAGES	TR-211
Bobrow, D.G.	NATURAL LANGUAGE INPUT FOR A COMPUTER PROBLEM SOLVING SYSTEM	TR-001
Bonneau, R. J.	AN INTERACTIVE IMPLEMENTATION OF THE TODD-COXETER ALGORITHM	TM-035
Bonneau, R. J.	POLYNOMIAL EXPONENTIATION: THE FAST FOURIER TRANSFORM REVISITED	TM-034
Bonneau, R.J.	A CLASS OF FINITE COMPUTATION STRUCTURES SUPPORTING THE FAST FOURIER ...	TM-031
Borkin, S.A.	DATA MODEL EQUIVALENCE	TM-118
Borkin, S.A.	EQUIVALENCE PROPERTIES OF SEMANTIC DATA MODELS FOR DATA BASE SYSTEMS	TR-206
Bosyj, M.	A PROGRAM FOR THE DESIGN OF PROCUREMENT SYSTEMS	TR-160
Brackett, J.	CASE STUDY IN INTERACTIVE GRAPHICS PROGRAMMING: A CIRCUIT DRAWING AND ...	TR-063
Brackett, J.	MAP: A SYSTEM FOR ON-LINE MATHEMATICAL ANALYSIS	TR-024
Bratt, R.G.	MINIMIZING THE NAMING FACILITIES REQUIRING PROTECTION IN A COMPUTING ...	TR-156
Briabrin, V.	AN ABSTRACT MODEL OF A RESEARCH INSTITUTE: SIMPLE AUTOMATIC PROGRAMMING ...	TR-118

Brock, J.D.	OPERATIONAL SEMANTICS OF A DATA FLOW LANGUAGE	TM-120
Bromley, Hank	A PROGRAM FOR THERAPY OF ACID-BASE AND ELECTROLYTE DISORDERS	TM-240
Brown, D.	STORAGE AND ACCESS COST FOR IMPLEMENTATIONS OF VARIABLE LENGTH LISTS	TR-217
Brown, G.P.	A SYSTEM TO PROCESS DIALOGUE: A PROGRESS REPORT	TM-079
Brown, G.P.	TOWARD A COMPUTATIONAL THEORY OF INDIRECT SPEECH ACTS	TR-223
Brown, G.P.	A FRAMEWORK FOR PROCESSING DIALOGUE	TR-182
Brown, G.P.	SOME PROBLEMS IN GERMAN TO ENGLISH MACHINE TRANSLATION	TR-142
Bruere-Dawson, G.	PSEUDO-RANDOM SEQUENCES	TM-016
Bruss, A.R.	ON TIME-SPACE CLASSES AND THEIR RELATION TO THE THEORY OF REAL ADDITION	TR-195
Bryant, R.E.	REPORT ON THE WORKSHOP ON SELF-TIMED SYSTEMS	TM-166
Bryant, R.E.	CONCURRENT PROGRAMMING	TM-115
Bryant, R.E.	A SWITCH-LEVEL SIMULATION MODEL FOR INTEGRATED LOGIC CIRCUITS	TR-259
Bryant, R.E.	SIMULATION OF PACKET COMMUNICATION ARCHITECTURE COMPUTER SYSTEMS	TR-188
Bui, T.N.	ON BISECTING RANDOM GRAPHS	TR-287
Burke, G.	MACLISP EXTENSIONS	TM-203
Burke, G.	LSB MANUAL	TM-200
Burke, G.	LOOP ITERATION MACRO	TM-169
Cardoza, E.W.	COMPUTATIONAL COMPLEXITY OF THE WORD PROBLEM FOR COMMUTATIVE SEMIGROUPS	TM-067
Cesari, C.A.	APPLICATION OF DATA FLOW ARCHITECTURE TO COMPUTER MUSIC SYNTHESIS	TR-257
Chan, A.Y.	INDEX SELECTION IN A SELF-ADAPTIVE RELATIONAL DATA BASE MANAGEMENT SYSTEM	TR-166
Chang, G.D.	DESIGN AND IMPLEMENTATION OF A TABLE-DRIVEN COMPILER SYSTEM	TR-042
Charniak, E.	CARPS, A PROGRAM WHICH SOLVES CALCULUS WORD PROBLEMS	TR-051
Cheek, T.B.	DESIGN OF A LOW-COST CHARACTER GENERATOR FOR REMOTE COMPUTER DISPLAYS	TR-026
Chmielinska, A.	HOARES'S LOGIC IS NOT COMPLETE WHEN IT COULD BE	TM-226
Chu, T.-A.	CIRCUIT ANALYSIS OF SELF-TIMED ELEMENTS FOR NMOS VLSI SYSTEMS	TM-220
Church, K.W.	COPING WITH SYNTACTIC AMBIGUITY OR HOW TO PUT THE BLOCK IN THE BOX ON THE ...	TM-216
Church, K.W.	PHRASE-STRUCTURE PARSING: A METHOD FOR TAKING ADVANTAGE OF ALLOPHONIC AND ...	TR-296
Church, K.W.	PRELIMINARY ANALYSIS OF A BREADTH-FIRST PARSING ALGORITHM: THEORETICAL ...	TR-261
Church, K.W.	ON MEMORY LIMITATIONS IN NATURAL LANGUAGE PROCESSING	TR-245
Clark, D.D., Editor	ANCILLARY REPORTS: KERNEL DESIGN PROJECT	TM-087
Clark, D.D.	DISTRIBUTED COMPUTER SYSTEMS: STRUCTURE AND SEMANTICS	TR-215
Clark, D.D.	FINAL REPORT OF THE MULTICS KERNEL DESIGN PROJECT	TR-196
Clark, D.D.	AN INPUT/OUTPUT ARCHITECTURE FOR VIRTUAL MEMORY COMPUTER SYSTEMS	TR-117
Clark, D.D.	THE CLASSROOM INFORMATION AND COMPUTING SERVICE	TR-080
Coons, S.A.	SURFACES FOR COMPUTER-AIDED DESIGN OF SPACE FORMS	TR-041
Corbató, F.J.	SYSTEM REQUIREMENTS FOR MULTIPLE ACCESS, TIME-SHARED COMPUTERS	TR-003
Cosmadakis, S.S.	THE COMPLEXITY OF EVALUATION RELATIONAL QUERIES	TM-229
Cosmadakis, S.S.	THE TRAVELING SALESMAN PROBLEM WITH MANY VISITS TO FEW CITIES	TM-208
Cosmadakis, S.S.	TRANSLATING UPDATES OF RELATIONAL DATA BASE VIEWS	TR-284
Davies, B.	A COMPUTER SYSTEM FOR DECISION ANALYSIS IN HODGKINS DISEASE	TR-271
Davis, E.	ALGORITHMS FOR SCHEDULING TASKS ON UNRELATED PROCESSORS	TM-137
Deital, H.M.	ABSENTEE COMPUTATIONS IN A MULTIPLE-ACCESS COMPUTER SYSTEM	TR-052
Denning, P.J.	RESOURCE ALLOCATION IN MULTIPROCESS COMPUTER SYSTEMS	TR-050
Denning, P.J.	QUEUEING MODELS FOR FILE MEMORY OPERATION	TR-021
Dennis, J.B.	CONCURRENT PROGRAMMING	TM-115
Dennis, J.B.	RESEARCH DIRECTIONS IN COMPUTER ARCHITECTURE	TM-114
Dennis, J.B.	FIRST VERSION OF A DATA FLOW PROCEDURE LANGUAGE	TM-061
Dennis, J.B.	VAL--A VALUE-ORIENTED ALGORITHMIC LANGUAGE, PRELIMINARY REFERENCE MANUAL	TR-218
Dennis, J.B.	ON THE DESIGN AND SPECIFICATION OF A COMMON BASE LANGUAGE	TR-101
Dennis, J.B.	PROGRAMMING SEMANTICS FOR MULTIPROGRAMMED COMPUTATIONS	TR-023
Dennis, J.B.	PROGRAM STRUCTURE IN A MULTI-ACCESS COMPUTER	TR-011
Deremer, F.L.	PRACTICAL TRANSLATORS FOR LR(K) LANGUAGES	TR-065
Desforges, J.F.	A COMPUTER SYSTEM FOR DECISION ANALYSIS IN HODGKINS DISEASE	TR-271
Desforges, J.F.	DIAGNOSTIC PLANNING AND CANCER MANAGEMENT	TR-169
diSessa, A.A.	A PRINCIPLED DESIGN FOR AN INTEGRATED COMPUTATIONAL ENVIRONMENT	TM-223
Dolev, D.	PLANAR EMBEDDING OF PLANAR GRAPHS	TM-237
dOliveira, C.R.	AN ANALYSIS OF COMPUTER DECENTRALIZATION	TM-090
Dornbrook, M.	THE MDL PROGRAMMING LANGUAGE PRIMER	TR-292
Edelberg, M.	INTEGRAL CONVEX POLYHEDRA AND AN APPROACH TO INTEGRALIZATION	TR-074
Edwards, D.J.	OCAS - ON-LINE CRYPTANALYTIC AID SYSTEM	TR-027
Ehrenfeucht, A.	PUMPING LEMMAS FOR REGULAR SETS	TM-170
Elias, P.	MINIMAX OPTIMAL UNIVERSAL CODEWORD SETS	TM-212
Ellis, D.J.	FORMAL SPECIFICATIONS FOR PACKET COMMUNICATION SYSTEMS	TR-189
Ellis, D.J.	SEMANTICS OF DATA STRUCTURES AND REFERENCES	TR-134
Estrin, D.L.	DATA COMMUNICATIONS VIA CABLE TELEVISION NETWORKS: TECHNICAL AND POLICY ...	TR-273
Evans, A. Jr.	THE BCPL REFERENCE MANUAL	TR-141
Even, S.	THE MAX FLOW ALGORITHM OF DINIC AND KARZANOV: AN EXPOSITION	TM-080
Fano, R.M.	THE MAC-SYSTEM: A PROGRESS REPORT	TR-012
Fateman, R.J.	ESSAYS IN ALGEBRAIC SIMPLIFICATION	TR-095
Faust, G.G.	SEMIAUTOMATIC TRANSLATION OF COBOL INTO HIBOL	TR-256
Feldman C.G.	VERBAL AND GRAPHICAL LANGUAGE FOR THE AED SYSTEM; A PROGRESS REPORT	TR-004
Fenichel, R.R.	A NEW LIST-TRACING ALGORITHM	TM-019
Fenichel, R.R.	AN ON-LINE SYSTEM FOR ALGEBRAIC MANIPULATION	TR-035
Ferrante, J.	A DECISION PROCEDURE FOR THE FIRST ORDER THEORY OF REAL ADDITION WITH ...	TM-033
Ferrante, J.	SOME UPPER AND LOWER BOUNDS ON DECISION PROCEDURES IN LOGIC	TR-139
Fiasconaro, J.G.	A COMPUTER-CONTROLLED GRAPHICAL DISPLAY PROCESSOR	TR-071

Fillat, A.I.	GENERALIZED ORGANIZATION OF LARGE DATA BASES: A SET-THEORETIC APPROACH TO ...	TR-070
Finseth, C.A.	THEORY AND PRACTICE OF TEXT EDITORS OR A COOKBOOK FOR AN EMACS	TM-165
Fischer, M.J.	Ω (n log n) LOWER BOUNDS ON LENGTH OF BOOLEAN FORMULAS	TM-187
Fischer, M.J.	A NOTE ON THE AVERAGE TIME TO COMPUTE TRANSITIVE CLOSURES	TM-076
Fischer, M.J.	THE COMPLEXITY OF NEGATION-LIMITED NETWORKS- A BRIEF SURVEY	TM-065
Fischer, M.J.	FAST ON-LINE INTEGER MULTIPLICATION	TM-045
Fischer, M.J.	SUPER-EXPONENTIAL COMPLEXITY OF PRESBURGER ARITHMETIC	TM-043
Fischer, M.J.	STRING-MATCHING AND OTHER PRODUCTS	TM-041
Fisher, M.J.	AN IMPROVED OVERLAP ARGUMENT FOR ON-LINE MULTIPLICATION	TM-040
Fischer, M.J.	IMPOSSIBILITY OF DISTRIBUTED CONSENSUS WITH ONE FAULTY PROCESS	TR-282
Forte, A.	SYNTAX-BASED ANALYTIC READING OF MUSICAL SCORES	TR-039
Frankston, R.M.	THE COMPUTER UTILITY AS A MARKETPLACE FOR COMPUTER SERVICES	TR-128
Fredkin, E.	CONSERVATIVE LOGIC	TM-197
Fuller, S.H.	RESEARCH DIRECTIONS IN COMPUTER ARCHITECTURE	TM-114
Furtek, F.C.	THE LOGIC OF SYSTEMS	TR-170
Galley, S.W.	THE MDL PROGRAMMING LANGUAGE	TR-293
Geiger, S. P.	A USER'S GUIDE TO THE MACRO CONTROL LANGUAGE	TM-036
Gertz, J.L.	HIERARCHICAL ASSOCIATIVE MEMORIES FOR PARALLEL COMPUTATION	TR-069
Gifford, D.	HARDWARE ESTIMATION OF A PROCESS' PRIMARY MEMORY REQUIREMENTS	TM-081
Goldberg, H.J.	A ROBUST ENVIRONMENT FOR PROGRAM DEVELOPMENT	TR-175
Goldman, B.	DEADLOCK DETECTION IN COMPUTER NETWORKS	TR-185
Goldstein, R.C.	HELPING PEOPLE THINK	TM-025
Goldstein, R.C.	THE MACAIMS DATA MANAGEMENT SYSTEM	TM-024
Goldstein, R.C.	THE SUBSTANTIVE USE OF COMPUTERS FOR INTELLECTUAL ACTIVITIES	TM-021
Good, M.	AN EASE OF USE EVALUATION OF AN INTEGRATED EDITOR AND FORMATTER	TR-266
Goodman, L.I.	COMPLEXITY MEASURES FOR PROGRAMMING LANGUAGES	TM-017
Goodman, N.	COORDINATION OF PARALLEL PROCESSES IN THE ACTOR MODEL OF COMPUTATION	TR-173
Goree, J.A.	INTERNAL CONSISTENCY OF A DISTRIBUTED TRANSACTION SYSTEM WITH ORPHAN ...	TR-286
Gorry, G.A.	RESEARCH ON EXPERT SYSTEMS	TM-056
Gorry, G.A.	A SYSTEM FOR COMPUTER-AIDED DIAGNOSIS	TR-044
Graham, R.M.	USE OF HIGH LEVEL LANGUAGES FOR SYSTEMS PROGRAMMING	TM-013
Graham, R.M.	FILE MANAGEMENT AND RELATED TOPICS	TM-012
Graham, R.M.	THE CLASSROOM INFORMATION AND COMPUTING SERVICE	TR-080
Greenbaum, H.J.	A SIMULATOR OF MULTIPLE INTERACTIVE USERS TO DRIVE A TIME-SHARED COMPUTER ...	TR-058
Greenberg, B.	AN EXPERIMENTAL ANALYSIS OF PROGRAM REFERENCE PATTERNS IN THE MULTICS ...	TR-127
Greenberger, M.	THE PRIORITY PROBLEM	TR-022
Greenberger, M.	A NEW METHODOLOGY FOR COMPUTER SIMULATION	TR-013
Greenberger, M.	THE OPS-1 MANUAL	TR-008
Greif, I.	COOPERATIVE OFFICE WORK, TELECONFERENCING AND CALENDAR MANAGEMENT: A .	TM-218
Greif, I.	PCAL: A PERSONAL CALENDAR	TM-213
Greif, I.	SOFTWARE FOR THE 'ROLES' PEOPLE PLAY	TM-210
Greif, I.	PROGRAMS FOR DISTRIBUTED COMPUTING. THE CALENDAR APPLICATION	TM-168
Greif, I.	SPECIFYING THE SEMANTICS OF WHILE-PROGRAMS: A TUTORIAL AND CRITIQUE OF A ...	TM-130
Greif, I.	INDUCTION IN PROOFS ABOUT PROGRAMS	TR-093
Grief, I.	SEMANTICS OF COMMUNICATING PARALLEL PROCESSES	TR-154
Griffith, A.K.	COMPUTER RECOGNITION OF PRISMATIC SOLIDS	TR-073
Grochow, J.M.	THE GRAPHIC DISPLAY AS AN AID IN THE MONITORING OF A TIME-SHARED COMPUTER ...	TR-054
Grossman, R.W.	SOME DATA BASE APPLICATIONS OF CONSTRAINT EXPRESSIONS	TR-158
Guang-Rong, G.	AN IMPLEMENTATION SCHEME FOR ARRAY OPERATIONS IN STATIC DATA FLOW ...	TR-280
Guttag, J.V.	DERIVED PAIRS, OVERLAP CLOSURES, AND REWRITE DOMINOES: NEW TOOLS FOR ...	TR-268
Guzman-Arenas, A.	COMPUTER RECOGNITION OF THREE-DIMENSIONAL OBJECTS IN A VISUAL SCENE	TR-059
Guzman-Arenas, A.	SOME ASPECTS OF PATTERN RECOGNITION BY COMPUTER	TR-037
Hack, M.	DECISION PROBLEMS FOR PETRI NETS AND VECTOR ADDITION SYSTEMS	TM-059
Hack, M.	DECIDABILITY QUESTIONS FOR PETRI NETS	TR-161
Hack, M.	PETRI NET LANGUAGES	TR-159
Hack, M.	ANALYSIS OF PRODUCTION SCHEMATA BY PETRI NETS	TR-094
Haggerty, J.P.	COMPLEXITY MEASURES FOR LANGUAGE RECOGNITION BY CANONIC SYSTEMS	TR-077
Halpern, J.Y.	ON THE EXPRESSIVE POWER OF DYNAMIC LOGIC, II	TM-204
Halpern, J.Y.	THE PROPOSITIONAL DYNAMIC LOGIC OF DETERMINISTIC, WELL-STRUCTURED PROGRAMS	TM-198
Halpern, J.Y.	DETERMINISTIC PROPOSITIONAL DYNAMIC LOGIC: FINITE MODELS, COMPLEXITY, AND ...	TM-190
Halpern, J.Y.	AXIOMATIC DEFINITIONS OF PROGRAMMING LANGUAGES, II	TM-179
Halpern, J.Y.	AXIOMATIC DEFINITIONS OF PROGRAMMING LANGUAGES: A THEORETICAL ASSESSMENT	TM-163
Halstead, R.H.	REFERENCE TREE NETWORKS: VIRTUAL MACHINE AND IMPLEMENTATION	TR-222
Halstead, R.H.	MULTIPLE-PROCESSOR IMPLEMENTATIONS OF MESSAGE PASSING SYSTEMS	TR-198
Hammer, M. M.	A NEW GRAMMATICAL TRANSFORMATION INTO DETERMINISTIC TOP-DOWN FORM	TR-119
Hammer, M.M.	ON DATA BASE MANAGEMENT SYSTEM ARCHITECTURE	TM-141
Hammer, M.M.	OFFICE ANALYSIS: METHODOLOGY AND CASE STUDIES	TR-289
Hammer, M.M.	CASE STUDY IN INTERACTIVE GRAPHICS PROGRAMMING: A CIRCUIT DRAWING AND ...	TR-063
Harel, D.	ARITHMETICAL COMPLETENESS IN LOGICS OF PROGRAMS	TM-103
Harel, D.	NONDETERMINISM IN LOGICS OF PROGRAMS	TM-098
Harel, D.	COMPUTABILITY AND COMPLETENESS IN LOGICS OF PROGRAMS	TM-097
Harel, D.	A COMPLETE AXIOMATIC SYSTEM FOR PROVING DEDUCTIONS ABOUT RECURSIVE ...	TM-096
Harel, D.	CHARACTERIZING SECOND ORDER LOGIC WITH FIRST ORDER QUANTIFIERS	TM-095
Harel, D.	LOGICS OF PROGRAMS: AXIOMATICS AND DESCRIPTIVE POWER	TR-200
Harper, L.H.	A CLASS OF BOOLEAN FUNCTIONS WITH LINEAR COMBINATIONAL COMPLEXITY	TM-055
Hawkinson, L.B.	XLMS: A LINGUISTIC MEMORY SYSTEM	TM-173

Hawkinson, L.B.	AN OVERVIEW OF OWL, A LANGUAGE FOR KNOWLEDGE REPRESENTATION	TM-086
Hawryszkiewycz, I.T.	SEMANTICS OF DATA BASE SYSTEMS	TR-112
Hebalkar, P.G.	DEADLOCK-FREE SHARING OF RESOURCES IN ASYNCHRONOUS SYSTEMS	TR-075
Henderson, D.A.	THE BINDING MODEL: A SEMANTIC BASE FOR MODULAR PROGRAMMING SYSTEMS	TR-145
Herlihy, M.P.	TRANSMITTING ABSTRACT VALUES IN MESSAGES	TR-234
Herrmann, P.P.	ON REDUCIBILITY AMONG COMBINATORIAL PROBLEMS	TR-113
Hewitt, C.	FOUNDATIONS FOR OFFICE SEMANTICS	TM-225
Hewitt, C.	A REAL TIME GARBAGE COLLECTOR THAT CAN RECOVER TEMPORARY STORAGE QUICKLY	TM-184
Hewitt, C.	ACTORS AND CONTINUOUS FUNCTIONALS	TR-194
Hoffman, C.W.	MACLISP EXTENSIONS	TM-203
Horn, B.K.P.	SHAPE FROM SHADING: A METHOD FOR OBTAINING THE SHAPE OFA SMOOTH OPAQUE ...	TR-079
Hossley, R.	THE EMPTINESS PROBLEM FOR AUTOMATA ON INFINITE TREES	TM-029
Hossley, R.	FINITE TREE AUTOMATA AND ω-AUTOMATA	TR-102
Hsieh W.N.	A CLASS OF BOOLEAN FUNCTIONS WITH LINEAR COMBINATIONAL COMPLEXITY	TM-055
Huber, A.H.	A MULTI-PROCESS DESIGN OF PAGING SYSTEM	TR-171
Humphrey, R.A.	A LOW-COST OUTPUT TERMINAL FOR TIME-SHARED COMPUTERS	TR-038
Hunt, D.H.	A CASE STUDY OF INTERMODULE DEPENDENCIES IN A VIRTUAL MEMORY SUBSYSTEM	TR-174
Iaezolla, G.G.	MODELING AND DECOMPOSITION OF INFORMATION SYSTEMS FOR PERFORMANCE ...	TM-026
Ilson, R.	AN INTEGRATED APPROACH TO FORMATTED DOCUMENT PRODUCTION	TR-253
Isaman, D.L.	DATA-STRUCTURING OPERATIONS IN CONCURRENT COMPUTATIONS	TR-224
Itai, A.	COVERING GRAPHS BY SIMPLE CIRCUITS	TM-155
Itai, A	HAMILTON PATHS IN GRID GRAPHS	TM-182
Ivie, E.L.	SEARCH PROCEDURES BASED ON MEASURES OF RELATEDNESS BETWEEN DOCUMENTS	TR-029
Jackson, J.N.	INTERACTIVE DESIGN COORDINATION FOR THE BUILDING INDUSTRY	TM-010
Jaffe, J.M.	ALGORITHMS FOR SCHEDULING TASKS ON UNRELATED PROCESSORS	TM-137
Jaffe, J.M.	EFFICIENT SCHEDULING OF TASKS WITHOUT FULL USE OF PROCESSOR RESOURCES	TM-122
Jaffe, J.M.	THE EQUIVALENCE OF R. E. PROGRAMS AND DATA FLOW SCHEMES	TM-121
Jaffe, J.M.	BOUNDS ON THE SCHEDULING OF TYPED TASK SYSTEMS	TM-111
Jaffe, J.M.	AN ANALYSIS OF PREEMPTIVE MULTIPROCESSOR JOB SCHEDULING	TM-110
Jaffe, J.M.	THE USE OF QUEUES IN THE PARALLEL DATA FLOW EVALUATION OF "IF-THEN-WHILE" ...	TM-104
Jaffe, J.M.	PARALLEL COMPUTATION: SYNCHRONIZATION, SCHEDULING, AND SCHEMES	TR-231
Jaffe, J.M.	SEMILINEAR SETS AND APPLICATIONS	TR-183
Janson, P.A.	USING TYPE EXTENSION TO ORGANIZE VIRTUAL MEMORY MECHANISMS	TR-167
Janson, P.A.	REMOVING THE DYNAMIC LINKER FROM THE SECURITY KERNEL OF A COMPUTING ...	TR-132
Johnson, D.	COMPUTATIONAL COMPLEXITY AND THE TRAVELING SALESMAN PROBLEM	TM-209
Johnson, D.	NEAR-OPTIMAL BIN PACKING ALGORITHMS	TR-109
Johnson, J.	PROGRAM RESTRUCTURING FOR VIRTUAL MEMORY SYSTEMS	TR-14
Johnson, M.G.	EFFICIENT MODELING FOR SHORT CHANNEL MOS CIRCUIT SIMULATION	TR-277
Jones, M.M.	INCREMENTAL SIMULATION ON A TIME-SHARED COMPUTER	TR-048
Jones, T.L.	A COMPUTER MODEL OF SIMPLE FORMS OF LEARNING	TM-020
Jouannaud, J.-P.	RECURSIVE DECOMPOSITION ORDERING AND MULTISET ORDERINGS	TM-219
Kahn, K.M.	MECHANIZATION OF TEMPORAL KNOWLEDGE	TR-155
Kaiser, G.E.	AUTOMATIC EXTENSION OF AN AUGMENTED TRANSITION NETWORK GRAMMAR FOR MORSE ...	TR-233
Kanellakis, P.C.	ON THE COMPUTATIONAL COMPLEXITY OF CARDINALITY CONSTRAINTS IN RELATIONAL ...	TM-160
Kanellakis, P.C.	THE COMPLEXITY OF CONCURRENCY CONTROL FOR DISTRIBUTED DATA BASES	TR-269
Kannon, R.	CIRCUIT-SIZE LOWER BOUNDS AND NON-REDUCIBILITY TO SPARSE SETS	TM-205
Kaplow, R.	MAP: A SYSTEM FOR ON-LINE MATHEMATICAL ANALYSIS	TR-024
Kapur, D.	DERIVED PAIRS, OVERLAP CLOSURES, AND REWRITE DOMINOES: NEW TOOLS FOR ...	TR-268
Kapur, D.	TOWARDS A THEORY FOR ABSTRACT DATA TYPES	TR-237
Karger, P.A.	NON-DISCRETIONARY ACCESS CONTROL FOR DECENTRALIZED COMPUTING SYSTEMS	TR-179
Karp, R.M.	ON LINEAR CHARACTERIZATIONS OF COMBINATORIAL OPTIMIZATION PROBLEMS	TM-154
Kathail, V.	A DATA FLOW ARCHITECTURE WITH TAGGED TOKENS	TM-174
Kedem, Z.M.	THE REDUCTION METHOD FOR ESTABLISHING LOWER BOUNDS ON THE NUMBER OF ...	TM-048
Kedem, Z.M.	COMBINING DIMENSIONALITY AND RATE OF GROWTH ARGUMENTS FOR ESTABLISHING ...	TM-046
Kennedy, D.W.	A LOW-COST OUTPUT TERMINAL FOR TIME-SHARED COMPUTERS	TR-038
Kent, S.	PROTECTING EXTERNALLY SUPPLIED SOFTWARE IN SMALL COMPUTERS	TR-255
Kent, S.	ENCRYPTION-BASED PROTECTION PROTOCOLS FOR INTERACTIVE USER-COMPUTER ...	TR-162
Kleitman, D.	AN ASYMPTOTICALLY OPTIMAL LAYOUT FOR THE SHUFFLE-EXCHANGE GRAPH	TM-231
Konopelski, L.J.	IMPLEMENTING INTERNET REMOTE LOGIN ON A PERSONAL COMPUTER	TM-233
Kosinski, P.R.	DENOTATIONAL SEMANTICS OF DETERMINATE AND NON-DETERMINATE DATA FLOW ...	TR-220
Krakauer, L.J.	COMPUTER ANALYSIS OF VISUAL PROPERTIES OF CURVED OBJECTS	TR-082
Kraning, L.A.	GENERALIZED ORGANIZATION OF LARGE DATA BASES: A SET-THEORETIC APPROACH TO ...	TR-070
Krizan, B. C.	A MINICOMPUTER NETWORK SIMULATION SYSTEM	TR-208
Kung, H.-T.	AN OPTIMALITY THEORY OF CONCURRENCY CONTROL FOR DATABASES	TM-185
Kunin, J.S.	OFFICE ANALYSIS: METHODOLOGY AND CASE STUDIES	TR-289
Kunin, J.S.	ANALYSIS AND SPECIFICATION OF OFFICE PROCEDURES	TR-275
Landau, S.E.	ON COMPUTING GALOIS GROUPS AND ITS APPLICATION TO SOLVABILITY BY RADICALS	TR-288
LaPaugh, A.S.	THE SUBGRAPH HOMEOMORPHISM PROBLEM	TM-099
LaPaugh, A.S.	ALGORITHMS FOR INTEGRATED CIRCUIT LAYOUT: AN ANALYTIC APPROACH	TR-248
Laventhal, M.S.	SYNTHESIS OF SYNCHRONIZATION CODE FOR DATA ABSTRACTIONS	TR-203
Laventhal, M.S.	VERIFICATION OF PROGRAMS OPERATING ON STRUCTURED DATA	TR-124
Leal-Cantu, N.	ON THE SIMULATION OF DYNAMIC SYSTEMS WITH LUD PARAMETERS AND TIME ...	TR-045
Lebling, P.D.	THEH MDL PROGRAMMING ENVIRONMENT	TR-294
Ledgard, H.F.	A FORMAL SYSTEM FOR DEFINING THE SYNTAX AND SEMANTICS OF COMPUTER ...	TR-060
Leighton, F.T.	AN APPROXIMATION ALGORITHM FOR MANHATTAN ROUTING	TM-238
Leighton, F.T.	PLANAR EMBEDDING OF PLANAR GRAPHS	TM-237

Leighton, F.T.	WAFER-SCALE INTEGRATION OF SYSTOLIC ARRAYS	TM-236
Leighton, F.T.	AN ASYMPTOTICALLY OPTIMAL LAYOUT FOR THE SHUFFLE-EXCHANGE GRAPH	TM-231
Leighton, F.T.	NEW LOWER BOUND TECHNIQUES FOR VLSI	TM-227
Leighton, F.T.	LAYOUTS FOR THE SHUFFLE-EXCHANGE GRAPH BASED ON THE COMPLEX PLANE DIAGRAM	TM-221
Leighton, F.T.	LAYOUTS FOR THE SHUFFLE-EXCHANGE GRAPH AND LOWER BOUND TECHNIQUES FOR VLSI	TR-274
Leiserson, C.E.	WAFER-SCALE INTEGRATION OF SYSTOLIC ARRAYS	TM-236
Leiserson, C.E.	OPTIMIZING SYNCHRONOUS SYSTEMS	TM-215
Leiserson, C.E.	OPTIMAL PLACEMENT FOR RIVER ROUTING	TM-206
Lepley, M.	AN ASYMPTOTICALLY OPTIMAL LAYOUT FOR THE SHUFFLE-EXCHANGE GRAPH	TM-231
Lepley, M.	LAYOUTS FOR THE SHUFFLE-EXCHANGE GRAPH BASED ON THE COMPLEX PLANE DIAGRAM	TM-221
Lescanne, P.	RECURSIVE DECOMPOSITION ORDERING AND MULTISET ORDERINGS	TM-219
Lester, B.P.	COST ANALYSIS OF DEBUGGING SYSTEMS	TR-090
Leth, J. W.	AN INTERMEDIATE FORM FOR DATA FLOW PROGRAMS	TM-143
Leung, C.K.C.	FORMAL PROPERTIES OF WELL-FORMED DATA FLOW SCHEMAS	TM-066
Leung, C.K.C.	FAULT TOLERANCE IN PACKET COMMUNICATION COMPUTER ARCHITECTURES	TR-250
Levin, L.A.	A CONCEPT OF INDEPENDENCE WITH APPLICATIONS IN VARIOUS FIELDS OF ...	TR-235
Levin, M.	ON BATESON'S LOGICAL LEVELS OF LEARNING	TM-057
Levin, M.	MATHEMATICAL LOGIC FOR COMPUTER SCIENTISTS	TR-131
Levine, P.H.	FACILITATING INTERPROCESS COMMUNICATION IN A HETEROGENEOUS NETWORK ...	TR-184
Lewin, D.	IN-PROCESS MANUFACTURING QUALITY CONTROL	TR-083
Lieberman, H.	A REAL TIME GARBAGE COLLECTOR THAT CAN RECOVER TEMPORARY STORAGE QUICKLY	TM-184
Lind, J.C.	COMPUTING IN LOGARITHMIC SPACE	TM-052
Linderman, J.P.	PRODUCTIVITY IN PARALLEL COMPUTATION SCHEMATA	TR-111
Lipski, W.	A FAST ALGORITHM FOR TESTING FOR SAFETY AND DETECTING DEADLOCKS IN LOCKED ...	TM-181
Lipski, W.	ON DATA BASES WITH INCOMPLETE INFORMATION	TM-142
Lipton, R.J.	COVERING GRAPHS BY SIMPLE CIRCUITS	TM-155
Liskov, B.	CLU REFERENCE MANUAL	TR-225
Liskov, B.	DISTRIBUTED COMPUTER SYSTEMS: STRUCTURE AND SEMANTICS	TR-215
Liu, C.L.	DESIGN AND IMPLEMENTATION OF A TABLE-DRIVEN COMPILER SYSTEM	TR-042
Lloyd, E.L.	CRITICAL PATH SCHEDULING OF TASK SYSTEMS WITH RESOURCE AND PROCESSOR ...	TM-161
Lloyd, E.L.	ON TRIANGULATIONS OF A SET OF POINTS IN THE PLANE	TM-088
Lloyd, E.L.	SCHEDULING TASK SYSTEMS WITH RESOURCES	TR-236
Logcher, R.D.	STRESS: A PROBLEM-ORIENTED LANGUAGE FOR STRUCTURAL ENGINEERING	TR-006
Long, W.J.	A PROGRAM WRITER	TR-187
Longo, G.	POWER SET MODELS OF LAMBDA-CALCULUS: THEORIES, EXPANSIONS, ISOMORPHISMS	TM-207
Longo, G.	RECURSION THEORETIC OPERATORS AND MORPHISMS ON NUMBERED SETS	TM-194
Loui, M.C.	ON TIME VERSUS SPACE III	TM-175
Loui, M.C.	SPACE-BOUNDED SIMULATION OF MULTITAPE TURING MACHINES	TM-148
Loui, M.C.	A SPACE BOUND FOR ONE-TAPE MULTIDIMENSIONAL TURING MACHINES	TM-145
Loui, M.C.	THE SPACE COMPLEXITY OF TWO PEBBLE GAMES ON TREES	TM-133
Loui, M.C.	MINIMUM REGISTER ALLOCATION IS COMPLETE IN POLYNOMIAL SPACE	TM-128
Loui, M.C.	SIMULATIONS AMONG MULTIDIMENSIONAL TURING MACHINES	TR-242
Luconi, F.L.	ASYNCHRONOUS COMPUTATIONAL STRUCTURES	TR-049
Luniewski, A.	THE ARCHITECTURE OF AN OBJECT BASED PERSONAL COMPUTER	TR-232
Luniewski, A.	A SIMPLE AND FLEXIBLE SYSTEM INITIALIZATION MECHANISM	TR-180
Lynch, N.A.	CONCURRENCY CONTROL FOR RESILIENT NESTED TRANSACTIONS	TR-285
Lynch, N.A.	IMPOSSIBILITY OF DISTRIBUTED CONSENSUS WITH ONE FAULTY PROCESS	TR-282
Lynch, N.A.	MULTILEVEL ATOMICITY - A NEW CORRECTNESS CRITERION FOR DATA BASE ...	TR-281
Lynch, N.A.	RELATIVIZATION OF THE THEORY OF COMPUTATIONAL COMPLEXITY	TR-099
Mabee, R.F.	THE BCPL REFERENCE MANUAL	TR-141
Madnick, S.E.	STORAGE HIERARCHY SYSTEMS	TR-107
Madnick, S.E.	DESIGN STRATEGIES FOR FILE SYSTEMS	TR-078
Malhotra, A.	DESIGN CRITERIA FOR A KNOWLEDGE-BASED ENGLISH LANGUAGE SYSTEM FOR ...	TR-146
Malvania, N.	THE DESIGN OF A MODULAR LABORATORY FOR CONTROL ROBOTICS	TM-074
Mandl, R.	FURTHER RESULTS ON HIERARCHIES OF CANONIC SYSTEMS	TR-100
Manning, F.	AUTOMATIC TEST, CONFIGURATION AND REPAIR OF CELLULAR ARRAYS	TR-151
Manning, F.	AUTONOMOUS, SYNCHRONOUS COUNTERS CONSTRUCTED ONLY OF J-K FLIP-FLOPS	TR-096
Marcum, A.M.	A MANAGER FOR NAMED, PERMANENT OBJECTS	TM-162
Mark, W.S.	THE REFORMULATION MODEL OF EXPERTISE	TR-172
Mark, W.S.	A MODEL-DEBUGGING SYSTEM	TR-125
Marks, R.E.	DESIGN AND IMPLEMENTATION OF A TABLE-DRIVEN COMPILER SYSTEM	TR-042
Martin, W.A.	BRAND X MANUAL	TM-186
Martin, W.A.	ROLES, CO-DESCRIPTORS AND THE FORMAL REPRESENTATION OF QUANTIFIED ENGLISH ...	TM-139
Martin, W.A.	DESCRIPTIONS AND THE SPECIALIZATION OF CONCEPTS	TM-101
Martin, W.A.	AN OVERVIEW OF OWL, A LANGUAGE FOR KNOWLEDGE REPRESENTATION	TM-086
Martin, W.A.	PRELIMINARY ANALYSIS OF A BREADTH-FIRST PARSING ALGORITHM: THEORETICAL ...	TR-261
Martin, W.A.	A VERY HIGH LEVEL LANGUAGE FOR BUSINESS DATA PROCESSING	TR-254
Martin, W.A.	SYMBOLIC MATHEMATICAL LABORATORY	TR-036
Masek, W.J.	A FASTER ALGORITHM COMPUTING STRING EDIT DISTANCES	TM-105
Mason, A.H.	A LAYERED VIRTUAL MEMORY MANAGER	TR-177
Mayr, E.W.	THE COMPLEXITY OF THE WORD PROBLEMS FOR COMMUTATIVE SEMIGROUPS AND ...	TM-199
Mayr, E.W.	PERSISTENCE OF VECTOR REPLACEMENT SYSTEMS IS DECIDABLE	TM-189
Mayr, E.W.	AN EFFECTIVE REPRESENTATION OF THE REACHABILITY SET OF PERSISTENT PETRI ...	TM-188
Mayr, E.W.	THE COMPLEXITY OF THE FINITE CONTAINMENT PROBLEM FOR PETRI NETS	TR-181
Mc Cray, W.M.	SIM360: A S/360 SIMULATOR	TM-030
McLeod, D.	ON DATA BASE MANAGEMENT SYSTEM ARCHITECTURE	TM-141

McLeod, D.	A SEMANTIC DATA BASE MODEL AND ITS ASSOCIATED STRUCTURED USER INTERFACE	TR-214
McLeod, D.	HIGH LEVEL EXPRESSION OF SEMANTIC INTEGRITY SPECIFICATIONS IN A ...	TR-165
Meier zu Sieker, F.	A TELEX GATEWAY FOR THE INTERNET	TM-222
Meldman, J.A.	A PRELIMINARY STUDY IN COMPUTER-AIDED LEGAL ANALYSIS	TR-157
Metcalfe, R.M.	PACKET COMMUNICATION	TR-114
Meyer, A. R.	WEAK MONADIC SECOND ORDER THEORY OF SUCCESSOR IS NOT ELEMENTARY- RECURSIVE	TM-038
Meyer, A.R.	TERMINATIONS FOR RECURSIVE PROGRAMS: COMPLETENESS AND AXIOMATIC ...	TM-214
Meyer, A.R.	A NOTE ON EQUIVALENCES AMONG LOGICS OF PROGRAMS	TM-211
Meyer, A.R.	WHAT IS A MODEL OF THE LAMBDA CALCULUS? EXPANDED VERSION	TM-201
Meyer, A.R.	THE COMPLEXITY OF THE WORD PROBLEMS FOR COMMUTATIVE SEMIGROUPS AND ...	TM-199
Meyer, A.R.	THE DEDUCIBILITY PROBLEM IN PROPOSITIONAL DYNAMIC LOGIC	TM-192
Meyer, A.R.	Ω (n log n) LOWER BOUNDS ON LENGTH OF BOOLEAN FORMULAS	TM-187
Meyer, A.R.	A NOTE ON THE LENGTH OF CRAIG'S INTERPOLANTS	TM-183
Meyer, A.R.	AXIOMATIC DEFINITIONS OF PROGRAMMING LANGUAGES, II	TM-179
Meyer, A.R.	ON TIME VERSUS SPACE III	TM-175
Meyer, A.R.	WHAT IS A MODEL OF THE LAMBDA CALCULUS?	TM-171
Meyer, A.R.	AXIOMATIC DEFINITIONS OF PROGRAMMING LANGUAGES: A THEORETICAL ASSESSMENT	TM-163
Meyer, A.R.	ON THE EXPRESSIVE POWER OF DYNAMIC LOGIC	TM-157
Meyer, A.R.	DEFINABILITY IN DYNAMIC LOGIC	TM-156
Meyer, A.R.	TEN THOUSAND AND ONE LOGICS OF PROGRAMMING	TM-150
Meyer, A.R.	SPECIFYING THE SEMANTICS OF WHILE-PROGRAMS: A TUTORIAL AND CRITIQUE OF A ...	TM-130
Meyer, A.R.	WITH WHAT FREQUENCY ARE APPARENTLY INTRACTABLE PROBLEMS DIFFICULT?	TM-126
Meyer, A.R.	COMPUTABILITY AND COMPLETENESS IN LOGICS OF PROGRAMS	TM-097
Meyer, A.R.	A NOTE ON THE AVERAGE TIME TO COMPUTE TRANSITIVE CLOSURES	TM-076
Meyer, A.R.	THE INHERENT COMPUTATION COMPLEXITY OF THEORIES OF ORDERED SETS: A BRIEF ...	TM-054
Meyer, A.R.	AN IMPROVED OVERLAP ARGUMENT FOR ON-LINE MULTIPLICATION	TM-040
Meyer, A.R.	DISCRETE COMPUTATION: THEORY AND OPEN PROBLEMS	TM-039
Miller, G.L.	AN ASYMPTOTICALLY OPTIMAL LAYOUT FOR THE SHUFFLE-EXCHANGE GRAPH	TM-231
Miller, G.L.	LAYOUTS FOR THE SHUFFLE-EXCHANGE GRAPH BASED ON THE COMPLEX PLANE DIAGRAM	TM-221
Miller, J.R.	ON-LINE ANALYSIS FOR SOCIAL SCIENTISTS	TR-040
Miller, P.B.	STRATEGY SELECTION IN MEDICAL DIAGNOSIS	TR-153
Miller, P.L.	AUTOMATIC CODE-GENERATION FROM AN OBJECT-MACHINE DESCRIPTION	TM-018
Miller, P.L.	AUTOMATIC CREATION OF A CODE GENERATOR FROM A MACHINE DESCRIPTION	TR-085
Mirkowska, G.	THE DEDUCIBILITY PROBLEM IN PROPOSITIONAL DYNAMIC LOGIC	TM-192
Misunas, D.P.	REPORT ON THE SECOND WORKSHOP ON DATA FLOW COMPUTER AND PROGRAM ...	TM-136
Misunas, D.P.	A COMPUTER ARCHITECTURE FOR DATA FLOW COMPUTATION	TM-100
Misunas,D.P.	REPORT ON THE WORKSHOP ON DATA FLOW COMPUTER AND PROGRAM ORGANIZATION	TM-092
Mitchell, J.C.	THE IMPLICATION PROBLEM FOR FUNCTIONAL AND INCLUSION DEPENDENCIES	TM-235
Mitchell, J.C.	TERMINATIONS FOR RECURSIVE PROGRAMS: COMPLETENESS AND AXIOMATIC ...	TM-214
Mok, A.K.	TASK SCHEDULING IN THE CONTROL ROBOTICS ENVIRONMENT	TM-077
Mok, A.K.	FUNDAMENTAL DESIGN PROBLEMS OF DISTRIBUTED SYSTEMS FOR THE HARD-REAL-TIME ...	TR-297
Moll R.	AN OPERATOR EMBEDDING THEOREM FOR COMPLEXITY CLASSES OF RECURSIVE ...	TM-032
Moll, R.	COMPLEXITY CLASSES OF RECURSIVE FUNCTIONS	TR-110
Montgomery, W.A.	ROBUST CONCURRENCY CONTROL FOR A DISTRIBUTED INFORMATION SYSTEM	TR-207
Montgomery, W.A.	A SECURE AND FLEXIBLE MODEL OF PROCESS INITIATION FOR A COMPUTER UTILITY	TR-163
Montz, L.B.	SAFETY AND OPTIMIZATION TRANSFORMATIONS FOR DATA FLOW PROGRAMS	TR-240
Moon, D.	LOOP ITERATION MACRO	TM-169
Morris, J.H.	LAMBDA CALCULUS MODELS OF PROGRAMMING LANGUAGES	TR-057
Moses, J.	SYMBOLIC INTEGRATION	TR-047
Moss, E.B.	NESTED TRANSACTIONS: AN APPROACH TO RELIABLE DISTRIBUTED COMPUTING	TR-260
Moss, E.B.	CLU REFERENCE MANUAL	TR-225
Moss, E.B.	ABSTRACT DATA TYPES IN STACK BASED LANGUAGES	TR-190
Musser, D.R.	DERIVED PAIRS, OVERLAP CLOSURES, AND REWRITE DOMINOES: NEW TOOLS FOR ...	TR-268
Myers, J. M.	ANALYSIS OF THE SIMPLE CODE FOR DATA FLOW COMPUTATION	TR-216
Niamir, B.	ATTRIBUTE PARTITIONING IN A SELF-ADAPTIVE RELATIONAL DATA BASE SYSTEM	TR-192
Norton, L.M.	ADEPT - A HEURISTIC PROGRAM FOR PROVING THEOREMS OF GROUP THEORY	TR-033
Owens, R.C.	PRIMARY ACCESS CONTROL IN LARGE-SCALE TIME-SHARED DECISION SYSTEMS	TR-089
Pallottino, S.	AN EFFICIENT ALGORITHM FOR DETERMINING THE LENGTH OF THE LONGEST DEAD ...	TM-149
Papadimitriou, C.H.	TWO REMARKS ON THE POWER OF COUNTING	TM-228
Papadimitriou, C.H.	COMPUTATIONAL COMPLEXITY AND THE TRAVELING SALESMAN PROBLEM	TM-209
Papadimitriou, C.H.	THE TRAVELING SALESMAN PROBLEM WITH MANY VISITS TO FEW CITIES	TM-208
Papadimitriou, C.H.	ALGEBRAIC DEPENDENCIES	TM-193
Papadimitriou, C.H.	AN OPTIMALITY THEORY OF CONCURRENCY CONTROL FOR DATABASES	TM-185
Papadimitriou, C.H.	HAMILTON PATHS IN GRID GRAPHS	TM-182
Papadimitriou, C.H.	A FAST ALGORITHM FOR TESTING FOR SAFETY AND DETECTING DEADLOCKS IN LOCKED ...	TM-181
Papadimitriou, C.H.	A THEOREM IN DATA BASE CONCURRENCY CONTROL	TM-180
Papadimitriou, C.H.	COVERING GRAPHS BY SIMPLE CIRCUITS	TM-155
Papadimitriou, C.H.	ON LINEAR CHARACTERIZATIONS OF COMBINATORIAL OPTIMIZATION PROBLEMS	TM-154
Papadimitriou, C.H.	WORST-CASE AND PROBABILISTIC ANALYSIS OF A GEOMETRIC LOCATION PROBLEM	TM-153
Papadimitriou, C.H.	ON THE COMPLEXITY OF INTEGER PROGRAMMING	TM-152
Papadimitriou, C.H.	SERIALIZABILITY OF CONCURRENT DATA BASE UPDATES	TR-210
Parikh, R.	PROPOSITIONAL DYNAMIC LOGICS OF PROGRAMS: A SURVEY	TM-191
Parikh, R.	PUMPING LEMMAS FOR REGULAR SETS	TM-170
Parikh, R.	DEFINABILITY IN DYNAMIC LOGIC	TM-156
Parikh, R.	A DECIDABILITY RESULT FOR A SECOND ORDER PROCESS LOGIC	TM-112
Parikh, R.	EFFECTIVENESS	TM-109

Parikh, R.	A COMPLETENESS RESULT FOR A PROPOSITIONAL DYNAMIC LOGIC	TM-106
Paseman, W.G.	SOME NEW METHODS OF MUSIC SYNTHESIS	TM-172
Paterson, M.S.	Ω (n log n) LOWER BOUNDS ON LENGTH OF BOOLEAN FORMULAS	TM-187
Paterson, M.S.	WITH WHAT FREQUENCY ARE APPARENTLY INTRACTABLE PROBLEMS DIFFICULT?	TM-126
Paterson, M.S.	A FASTER ALGORITHM COMPUTING STRING EDIT DISTANCES	TM-105
Paterson, M.S.	STRING-MATCHING AND OTHER PRODUCTS	TM-041
Paterson, M.S.	IMPOSSIBILITY OF DISTRIBUTED CONSENSUS WITH ONE FAULTY PROCESS	TR-282
Paterson, M.S	AN IMPROVED OVERLAP ARGUMENT FOR ON-LINE MULTIPLICATION	TM-040
Patil, R.S.	COPING WITH SYNTACTIC AMBIGUITY OR HOW TO PUT THE BLOCK IN THE BOX ON THE ...	TM-216
Patil, R.S.	DESIGN OF A PROGRAM FOR EXPERT DIAGNOSIS OF ACID BASE AND ELECTROLYTE ...	TM-132
Patil,R.S.	CAUSAL REPRESENTATION OF PATIENT ILLNESS FOR ELECTROLYTE AND ACID-BASE ...	TR-267
Patil, R.S.	PRELIMINARY ANALYSIS OF A BREADTH-FIRST PARSING ALGORITHM: THEORETICAL ...	TR-261
Patil, S.S.	AN ASYNCHRONOUS LOGIC ARRAY	TM-062
Patil, S.S.	COORDINATION OF ASYNCHRONOUS EVENTS	TR-072
Pavelle, R.	COMPUTER PROGRAMS FOR RESEARCH IN GRAVITATION AND DIFFERENTIAL GEOMETRY	TM-167
Perry, H.M.	AN IMPROVED PROOF OF THE RABIN-HARTMANIS-STEARNS CONJECTURE	TM-123
Pfister, G.F.	THE COMPUTER CONTROL OF CHANGING PICTURES	TR-135
Pfister, G.	THE MDL PROGRAMMING LANGUAGE	TR-293
Pingali, K.	A DATA FLOW ARCHITECTURE WITH TAGGED TOKENS	TM-174
Pinter, R.Y.	OPTIMAL PLACEMENT FOR RIVER ROUTING	TM-206
Pinter, R.Y.	THE IMPACT OF LAYER ASSIGNMENT METHODS ON LAYOUT ALGORITHMS FOR ...	TR-291
Pitman, K.M.	THE REVISED MACLISP MANUAL	TR-295
Pless, V.	ENCRYPTION SCHEMES FOR COMPUTER CONFIDENTIALITY	TM-063
Pless, V.	COMPLETE CLASSIFICATION OF (24,12) AND (22,11) SELF-DUAL CODES	TM-049
Pless, V.	MATHEMATICAL FOUNDATIONS OF FLIP-FLOPS	TM-047
Pless, V.	SYMMETRY CODES AND THEIR INVARIANT SUBCODES	TM-044
Pnueli, A.	DETERMINISTIC PROPOSITIONAL DYNAMIC LOGIC: FINITE MODELS, COMPLEXITY, AND ..	TM-190
Pnueli, A.	A COMPLETE AXIOMATIC SYSTEM FOR PROVING DEDUCTIONS ABOUT RECURSIVE ...	TM-096
Pratt, V.R.	DYNAMIC ALGEBRAS AND THE NATURE OF INDUCTION	TM-159
Pratt, V.R.	DYNAMIC ALGEBRAS: EXAMPLES, CONSTRUCTIONS, APPLICATIONS	TM-138
Pratt, V.R.	SIX LECTURES ON DYNAMIC LOGIC	TM-117
Pratt, V.R.	APPLICATIONS OF MODAL LOGIC TO PROGRAMMING	TM-116
Pratt, V.R.	A NEAR-OPTIMAL METHOD FOR REASONING ABOUT ACTION	TM-113
Pratt, V.R.	NONDETERMINISM IN LOGICS OF PROGRAMS	TM-098
Pratt, V.R.	COMPUTABILITY AND COMPLETENESS IN LOGICS OF PROGRAMS	TM-097
Pratt, V.R.	THE MUTUAL EXCLUSION PROBLEM FOR UNRELIABLE PROCESSES	TM-084
Pratt, V.R.	SEMANTICAL CONSIDERATIONS ON FLOYD-HOARE LOGIC	TR-168
Principato, R.N.	A FORMALIZATION OF THE STATE MACHINE SPECIFICATION TECHNIQUE	TR-202
Qualitz, J.E.	DECIDABILITY OF EQUIVALENCE FOR A CLASS OF DATA FLOW SCHEMAS	TM-058
Qualitz, J.E.	EQUIVALENCE PROBLEMS FOR MONADIC SCHEMAS	TR-152
Rabin, M.O.	SUPER-EXPONENTIAL COMPLEXITY OF PRESBURGER ARITHMETIC	TM-043
Rabin, M.O.	PROBABILISTIC ALGORITHMS IN FINITE FIELDS	TR-213
Rabin, M.O.	DIGITALIZED SIGNATURES AND PUBLIC-KEY FUNCTIONS AS INTRACTABLE AS ...	TR-212
Rackoff, C.	ON THE COMPLEXITY OF THE THEORIES OF WEAK DIRECT PRODUCTS	TM-042
Rackoff, C.	A DECISION PROCEDURE FOR THE FIRST ORDER THEORY OF REAL ADDITION WITH ...	TM-033
Rackoff, C.	THE EMPTINESS PROBLEM FOR AUTOMATA ON INFINITE TREES	TM-029
Rackoff, C.	THE COMPUTATIONAL COMPLEXITY OF SOME LOGICAL THEORIES	TR-144
Rackoff, C.	THE EMPTINESS AND COMPLEMENTATION PROBLEMS FOR AUTOMATA ON INFINITE TREES	TR-106
Ramchandani, C.	ANALYSIS OF ASYNCHRONOUS CONCURRENT SYSTEMS BY TIMED PETRI NETS	TR-120
Raphael, B.	SIR: A COMPUTER PROGRAM FOR SEMANTIC INFORMATION RETRIEVAL	TR-002
Rappaport, R.L.	IMPLEMENTING MULTI-PROCESS PRIMITIVES IN A MULTIPLEXED COMPUTER SYSTEM	TR-055
Redell, D.D.	NAMING AND PROTECTION IN EXTENDABLE OPERATING SYSTEMS	TR-140
Reed, D.P.	NAMING AND SYNCHRONIZATION IN A DECENTRALIZED COMPUTER SYSTEMS	TR-205
Reed, D.P.	PROCESSOR MULTIPLEXING IN A LAYERED OPERATING SYSTEM	TR-164
Reif, J.H.	THE PROPOSITIONAL DYNAMIC LOGIC OF DETERMINISTIC, WELL-STRUCTURED PROGRAMS	TM-198
Reinig, F.	RECURSIVE DECOMPOSITION ORDERING AND MULTISET ORDERINGS	TM-219
Reuveni, A.	THE EVENT BASED LANGUAGE AND ITS MULTIPLE PROCESSOR IMPLEMENTATIONS	TR-226
Richards, M.	THE BCPL REFERENCE MANUAL	TR-141
Rivest, R.	RANDOMIZED ENCRYPTION TECHNIQUES	TM-234
Rivest, R.	MENTAL POKER	TM-125
Rivest, R.	THE MUTUAL EXCLUSION PROBLEM FOR UNRELIABLE PROCESSES	TM-084
Rivest, R.	A METHOD FOR OBTAINING SIGNATURES AND PUBLIC-KEY CRYPTOSYSTEMS	TM-082
Rivest, R.	K + 1 HEADS ARE BETTER THAN K	TM-075
Rivest, R.	OPTIMAL ARRANGEMENT OF KEYS IN A HASH TABLE	TM-073
Rivest, R.	ON THE WORST-CASE OF BEHAVIOR OF STRING-SEARCHING ALGORITHMS	TM-071
Rodeh, M.	COVERING GRAPHS BY SIMPLE CIRCUITS	TM-155
Rodrigues, J.E.	A GRAPH MODEL FOR PARALLEL COMPUTATIONS	TR-064
Rodriguez, H.	MEASURING USER CHARACTERISTICS ON THE MULTICS SYSTEM	TM-089
Roos, D.	USE OF CTSS IN A TEACHING ENVIRONMENT	TR-014
Rosenberg, R.L.	INCOMPREHENSIBLE COMPUTER SYSTEMS: KNOWLEDGE WITHOUT WISDOM	TR-227
Rosenburg, R.C.	A LOW-COST OUTPUT TERMINAL FOR TIME-SHARED COMPUTERS	TR-038
Rosenstein, L.S.	DISPLAY MANAGEMENT IN AN INTEGRATED OFFICE	TR-278
Ross, D.T.	AN INTEGRATED HARDWARE-SOFTWARE SYSTEM FOR COMPUTER GRAPHICS IN .	TR-056
Ross, D.T.	VERBAL AND GRAPHICAL LANGUAGE FOR THE AED SYSTEM; A PROGRESS REPORT	TR-004
Rotenberg, L.	MAKING COMPUTERS KEEP SECRETS	TR-115
Rozenberg, G.	PUMPING LEMMAS FOR REGULAR SETS	TM-170

Rumbaugh, J.E.	A PARALLEL ASYNCHRONOUS COMPUTER ARCHITECTURE FOR DATA FLOW PROGRAMS	TR-150
Russo, F.J.	A HEURISTIC APPROACH TO ALTERNATE ROUTING IN A JOB SHOP	TR-019
Ruth, G.R.	PROTOSYSTEM I: AN AUTOMATIC PROGRAMMING SYSTEM PROTOTYPE	TM-072
Ruth, G.R.	AUTOMATIC DESIGN OF DATA PROCESSING SYSTEMS	TM-070
Ruth, G.R.	A VERY HIGH LEVEL LANGUAGE FOR BUSINESS DATA PROCESSING	TR-254
Ruth, G.R.	DATA DRIVEN LOOPS	TR-244
Ruth, G.R.	ANALYSIS OF ALGORITHM IMPLEMENTATIONS	TR-130
Rutherford, C.J.	A COMPUTER SYSTEM FOR DECISION ANALYSIS IN HODGKINS DISEASE	TR-271
Safran, C.	DIAGNOSTIC PLANNING AND CANCER MANAGEMENT	TR-169
Saltzer, et al.	INTRODUCTION TO MULTICS	TR-123
Saltzer, J.H.	COMMUNICATION RING INITIALIZATION WITHOUT CENTRAL CONTROL	TM-202
Saltzer, J.H.	FINAL REPORT OF THE MULTICS KERNEL DESIGN PROJECT	TR-196
Saltzer, J.H.	THE CLASSROOM INFORMATION AND COMPUTING SERVICE	TR-080
Saltzer, J.H.	TRAFFIC CONTROL IN A MULTIPLEXED COMPUTER	TR-030
Saltzer, J.H.	CTSS TECHNICAL NOTES	TR-016
Samuel, A.L.	TIME-SHARING ON A MULTICONSOLE COMPUTER	TR-017
Savage. J.E.	A CLASS OF BOOLEAN FUNCTIONS WITH LINEAR COMBINATIONAL COMPLEXITY	TM-055
Saxe, J.B.	OPTIMIZING SYNCHRONOUS SYSTEMS	TM-215
Schaffert, C.	CLU REFERENCE MANUAL	TR-225
Schaffert, C.	A FORMAL DEFINITION OF CLU	TR-193
Scheifler, R.	CLU REFERENCE MANUAL	TR-225
Scheifler, R.	A DENOTATIONAL SEMANTICS OF CLU	TR-201
Schell, R.R.	DYNAMIC RECONFIGURATION IN A MODULAR COMPUTER SYSTEM	TR-086
Scherf, J.A.	COMPUTER AND DATA SECURITY: A COMPREHENSIVE ANNOTATED BIBLIOGRAPHY	TR-122
Scherr, A.L.	AN ANALYSIS OF TIME-SHARED COMPUTER SYSTEMS	TR-018
Schiffenbauer, R.D.	INTERACTIVE DEBUGGING IN A DISTRIBUTED COMPUTATIONAL ENVIRONMENT	TR-264
Schoichet, S.R.	OFFICE ANALYSIS: METHODOLOGY AND CASE STUDIES	TR-289
Schonage, A.	REAL-TIME SIMULATION OF MULTIDIMENSIONAL TURING MACHINES BY STORAGE ...	TM-037
Schroeder, M.D.	FINAL REPORT OF THE MULTICS KERNEL DESIGN PROJECT	TR-196
Schroeder, M.D.	COOPERATION OF MUTUALLY SUSPICIOUS SUBSYSTEMS IN A COMPUTER UTILITY	TR-104
Schroeder, M.D.	THE CLASSROOM INFORMATION AND COMPUTING SERVICE	TR-080
Schroeppel, R.	A T = $0(2^{n/2})$, S = $0(2^{n/4})$ ALGORITHM FOR CERTAIN NP-COMPLETE PROBLEMS	TM-147
Seaquist, C.R.	A SEMANTICS OF SYNCHRONIZATION	TM-176
Seiferas, J.	NONDETERMINISTIC TIME AND SPACE COMPLEXITY CLASSES	TR-137
Sekino, A.	PERFORMANCE EVALUATION OF MULTIPROGRAMMED TIME-SHARED COMPUTER SYSTEMS	TR-103
Selwyn, L.	ECONOMIES OF SCALE IN COMPUTER USE: INITIAL TESTS AND IMPLICATIONS FOR ...	TR-068
Shamir, A.	EMBEDDING CRYPTOGRAPHIC TRAPDOORS IN ARBITRARY KNAPSACK SYSTEMS	TM-230
Shamir, A.	THE CRYPTOGRAPHIC SECURITY OF COMPACT KNAPSACKS-(PRELIMINARY REPORT	TM-164
Shamir, A.	A T = $0(2^{n/2})$, S = $0(2^{n/4})$ ALGORITHM FOR CERTAIN NP-COMPLETE PROBLEMS	TM-147
Shamir, A.	HOW TO SHARE A SECRET	TM-134
Shamir, A.	ON THE CRYPTOCOMPLEXITY OF KNAPSACK SYSTEMS	TM-129
Shamir, A.	MENTAL POKER	TM-125
Shamir, A.	ON THE SECURITY OF THE MERKLE-HELLMAN CRYPTOGRAPHIC SCHEME	TM-119
Shamir, A.	A FAST SIGNATURE SCHEME	TM-107
Shamir, A.	FACTORING NUMBERS IN 0 (LOG n) ARITHMETIC STEPS	TM-091
Shamir, A.	FINDING MINIMUM CUTSETS IN REDUCIBLE GRAPHS	TM-085
Shamir, A.	A METHOD FOR OBTAINING SIGNATURES AND PUBLIC-KEY CRYPTOSYSTEMS	TM-082
Sherman, A.T.	RANDOMIZED ENCRYPTION TECHNIQUES	TM-234
Sherman, H.B.	A COMPARATIVE STUDY OF COMPUTER-AIDED CLINICAL DIAGNOSIS	TR-283
Silverman, H.	A DIGITALIS THERAPY ADVISOR	TR-143
Singh, N.P.	A DESIGN METHODOLOGY FOR SELF-TIMED SYSTEMS	TR-258
Singh, V.	THE DESIGN OF A ROUTING SERVICE FOR CAMPUS-WIDE INTERNET TRANSPORT	TR-270
Sinha, M.K.	TIMEPAD - A PERFORMANCE IMPROVING SYNCHRONIZATION MECHANISM FOR ...	TM-177
Sirbu, M.	EVALUATION OF AN OFFICE ANALYSIS METHODOLOGY	TM-239
Sirbu, M.	OFFICE ANALYSIS: METHODOLOGY AND CASE STUDIES	TR-289
Sloane, N.J.A.	COMPLETE CLASSIFICATION OF (24,12) AND (22,11) SELF-DUAL CODES	TM-049
Slutz, D.R.	THE FLOW GRAPH SCHEMATA MODEL OF PARALLEL COMPUTATION	TR-053
Smith, A.A.	INPUT/OUTPUT IN TIME-SHARED, SEGMENTED, MULTIPROCESSOR SYSTEMS	TR-028
Smith, B.C.	REFLECTION AND SEMANTICS IN A PROCEDURAL LANGUAGE	TR-272
Smith, B.J.	AN ANALYSIS OF SORTING NETWORKS	TR-105
Smith, D.L.	MODELS AND DATA STRUCTURES FOR DIGITAL LOGIC SIMULATION	TR-031
Smoliar, S.W.	A PARALLEL PROCESSING MODEL OF MUSICAL STRUCTURES	TR-091
Snyder, A.	CLU REFERENCE MANUAL	TR-225
Snyder, A.	A MACHINE ARCHITECTURE TO SUPPORT AN OBJECT-ORIENTED LANGUAGE	TR-209
Snyder, A.	A PORTABLE COMPILER FOR THE LANGUAGE C	TR-149
Sollins, K.R.	COPYING COMPLEX STRUCTURES IN A DISTRIBUTED SYSTEM	TR-219
Srivas, M.K.	AUTOMATIC SYNTHESIS OF IMPLEMENTATIONS FOR ABSTRACT DATA TYPES FROM ...	TR-276
Stark, E.W.	SEMAPHORE PRIMITIVES AND STARVATION-FREE MUTUAL EXCLUSION	TM-158
Stavi, J.	A COMPLETE AXIOMATIC SYSTEM FOR PROVING DEDUCTIONS ABOUT RECURSIVE ...	TM-096
Stern, J.A.	BACKUP AND RECOVERY OF ON-LINE INFORMATION IN A COMPUTER UTILITY	TR-116
Stockmeyer, L.J.	FAST ON-LINE INTEGER MULTIPLICATION	TM-045
Stockmeyer, L.J.	THE COMPLEXITY OF DECISION PROBLEMS IN AUTOMATA THEORY AND LOGIC	TR-133
Stockmeyer, L.J.	BOUNDS ON POLYNOMIAL EVALUATION ALGORITHMS	TR-098
Stotz, R.H.	AN INTEGRATED HARDWARE-SOFTWARE SYSTEM FOR COMPUTER GRAPHICS IN . .	TR-056
Stratton, W.D.	INVESTIGATION OF AN ANALOG TECHNIQUE TO DECREASE PEN-TRACKING TIME IN ...	TR-025
Strazdas, R.J.	A NETWORK TRAFFIC GENERATOR FOR DECNET	TM-127

Street, R.S.	PROPOSITIONAL DYNAMIC LOGIC OF LOOPING AND CONVERSE	TR-263
Streett, R.S.	THE DEDUCIBILITY PROBLEM IN PROPOSITIONAL DYNAMIC LOGIC	TM-192
Strnad, A.J.	THE MACAIMS DATA MANAGEMENT SYSTEM	TM-024
Strnad, A.J.	THE RELATIONAL APPROACH TO THE MANAGEMENT OF DATA BASES	TM-023
Strong, S.	MAP: A SYSTEM FOR ON-LINE MATHEMATICAL ANALYSIS	TR-024
Sutherland, J.B.	EVALUATION OF AN OFFICE ANALYSIS METHODOLOGY	TM-239
Sutherland, J.B.	AN OFFICE ANALYSIS AND DIAGNOSIS METHODOLOGY	TR-290
Sutherland, J.B.	OFFICE ANALYSIS: METHODOLOGY AND CASE STUDIES	TR-289
Svobodova, L.	MANAGEMENT OF OBJECT HISTORIES IN THE SWALLOW REPOSITORY	TR-243
Svobodova, L.	DISTRIBUTED COMPUTER SYSTEMS: STRUCTURE AND SEMANTICS	TR-215
Swan, R.J.	RESEARCH DIRECTIONS IN COMPUTER ARCHITECTURE	TM-114
Swartout, W.R.	PRODUCING EXPLANATIONS AND JUSTIFICATIONS OF EXPERT CONSULTING PROGRAMS	TR-251
Swartout, W.R.	A DIGITALIS THERAPY ADVISOR WITH EXPLANATIONS	TR-176
Szolovits, P.	BRAND X MANUAL	TM-186
Szolovits, P.	ARTIFICIAL INTELLIGENCE AND CLINICAL PROBLEM SOLVING	TM-140
Szolovits, P.	AN OVERVIEW OF OWL, A LANGUAGE FOR KNOWLEDGE REPRESENTATION	TM-086
Szwarefiter, J.L.	HAMILTON PATHS IN GRID GRAPHS	TM-182
Takagi, A.	CONCURRENT AND RELIABLE UPDATES OF DISTRIBUTED DATA BASE	TM-144
Teitelman, W.	PILOT: A STEP TOWARDS MAN-COMPUTER SYMBIOSIS	TR-032
Teixeira, T.J.	REAL-TIME CONTROL STRUCTURES FOR BLOCK DIAGRAM SCHEMATA	TR-204
Terman, C.J.	THE SPECIFICATION OF CODE GENERATION ALGORITHMS	TR-199
Thomas, R.E.	I-STRUCTURES: AN EFFICIENT DATA TYPE FOR FUNCTIONAL LANGUAGES	TM-178
Thomas, R.E.	A DATA FLOW ARCHITECTURE WITH IMPROVED ASYMPTOTIC PERFORMANCE	TR-265
Thomas, R.H.	A MODEL FOR PROCESS REPRESENTATION AND SYNTHESIS	TR-087
Thornhill, D.	CASE STUDY IN INTERACTIVE GRAPHICS PROGRAMMING: A CIRCUIT DRAWING AND ...	TR-063
Thornhill, D.	AN INTEGRATED HARDWARE-SOFTWARE SYSTEM FOR COMPUTER GRAPHICS IN ...	TR-056
Tillman, C.C.	EPS: AN INTERACTIVE SYSTEM FOR SOLVING ELLIPTIC BOUNDARY-VALUE PROBLEMS ...	TR-062
Tiuryn, J.	HOARES'S LOGIC IS NOT COMPLETE WHEN IT COULD BE	TM-226
Tiuryn, J.	A NOTE ON EQUIVALENCES AMONG LOGICS OF PROGRAMS	TM-211
Tiuryn, J.	A SURVEY OF THE LOGIC OF EFFECTIVE DEFINITIONS	TR-246
Todd, K.W.	HIGH LEVEL VAL CONSTRUCTS IN A STATIC DATA FLOW MACHINE	TR-262
Toffoli, T.	CONSERVATIVE LOGIC	TM-197
Toffoli, T.	REVERSIBLE COMPUTING	TM-151
Toffoli, T.	AN EFFICIENT ALGORITHM FOR DETERMINING THE LENGTH OF THE LONGEST DEAD ...	TM-149
Toffoli, T.	BICONTINUOUS EXTENSIONS OF INVERTIBLE COMBINATORIAL FUNCTIONS	TM-124
Trickey, H.	PLANAR EMBEDDING OF PLANAR GRAPHS	TM-237
Tsichlis, P.N.	DIAGNOSTIC PLANNING AND CANCER MANAGEMENT	TR-169
Turkle, S.	COMPUTERS AND PEOPLE: PERSONAL COMPUTATION	TR-249
Van De Vanter, M.L.	A FORMALIZATION AND CORRECTNESS PROOF OF THE CGOL LANGUAGE SYSTEM	TR-147
Van Horn, E.C.	COMPUTER DESIGN FOR ASYNCHRONOUSLY REPRODUCIBLE MULTIPROCESSING	TR-034
Van Horn, E.C.	PROGRAMMING SEMANTICS FOR MULTIPROGRAMMED COMPUTATIONS	TR-023
Vanderbilt, D.H.	CONTROLLED INFORMATION SHARING IN A COMPUTER UTILITY	TR-067
Vilfan, B.	THE COMPLEXITY OF FINITE FUNCTIONS	TR-097
Vogt, C.M.	SUSPENSION OF PROCESSES IN A MULTIPROCESSING COMPUTER SYSTEM	TM-014
Wand, M.	MATHEMATICAL FOUNDATIONS OF FORMAL LANGUAGE THEORY	TR-108
Wang, P.S.	EVALUATION OF DEFINITE INTEGRALS BY SYMBOLIC MANIPULATION	TR-092
Wantman, M.E.	CALCULAID: AN ON-LINE SYSTEM FOR ALGEBRAIC COMPUTATION AND ANALYSIS	TR-020
Ward, J.E.	AN INTEGRATED HARDWARE-SOFTWARE SYSTEM FOR COMPUTER GRAPHICS IN ...	TR-056
Ward, P.W.	DESCRIPTION AND FLOW CHART OF THE PDP-7/9 COMMUNICATIONS PACKAGE	TM-011
Ward, S.	FUNCTIONAL DOMAINS OF APPLICATIVE LANGUAGES	TR-136
Weihl, W.E.	INTERPROCEDURAL DATA FLOW ANALYSIS IN THE PRESENCE OF POINTERS, PROCEDURE ...	TR-247
Weise, D.	ON TIME VERSUS SPACE III	TM-175
Weiss, R.B.	FINDING ISOMORPH CLASSES FOR COMBINATORIAL STRUCTURES	TM-064
Weiss, R.B.	CAMAC: GROUP MANIPULATION SYSTEM	TM-060
Weissberg, R.	USING INTERACTIVE GRAPHICS IN SIMULATING THE HOSPITAL EMERGENCY ROOM	TR-129
Weizenbaum, J.	OPL-I AN OPEN ENDED PROGRAMMING SYSTEM WITHIN CTSS	TR-007
Welch, T.A.	BOUNDS ON INFORMATION RETRIEVAL EFFICIENCY IN STATIC FILE STRUCTURES	TR-088
Wells, D.	TRANSMISSION OF INFORMATION BETWEEN A MAN-MACHINE DECISION SYSTEM AND ITS ...	TM-022
Wells, D.	FINAL REPORT OF THE MULTICS KERNEL DESIGN PROJECT	TR-196
Weng, K.-S.	RESEARCH DIRECTIONS IN COMPUTER ARCHITECTURE	TM-114
Weng, K.-S.	STREAM-ORIENTED COMPUTATION IN RECURSIVE DATA FLOW SCHEMAS	TM-068
Weng, K.-S.	AN ABSTRACT IMPLEMENTATION FOR A GENERALIZED DATA FLOW LANGUAGE	TR-228
Wester, M.	COMPUTER PROGRAMS FOR RESEARCH IN GRAVITATION AND DIFFERENTIAL GEOMETRY	TM-167
Wilde, D.U.	PROGRAM ANALYSIS BY DIGITAL COMPUTER	TR-043
Winklmann, K.	ON THE EXPRESSIVE POWER OF DYNAMIC LOGIC	TM-157
Winograd, T.	PROCEDURES AS A REPRESENTATION FOR DATA IN A COMPUTER PROGRAM FOR ...	TR-084
Winston, P.H.	LEARNING STRUCTURAL DESCRIPTION FROM EXAMPLES	TR-076
Wong, R.	CONSTRUCTION HEURISTICS FOR GEOMETRY AND A VECTOR ALGEBRA REPRESENTATION	TM-028
Wright, K.D.	A FILE TRANSFER PROGRAM FOR A PERSONAL COMPUTER	TM-217
Wyleczuk, R.H.	TIMESTAMPS AND CAPABILITY-BASED PROTECTION IN A DISTRIBUTED COMPUTER ...	TM-135
Yannakakis, M	ALGEBRAIC DEPENDENCIES	TM-193
Yao, A.C.	K+1 HEADS ARE BETTER THAN K	TM-075
Yao, F.F.	ON LOWER BOUNDS FOR SELECTION PROBLEMS	TR-121
Yeh, A.	PLY: A SYSTEM OF PLAUSIBILITY INFERENCE WITH A PROBABILISTIC BASIS	TM-232
Yonezawa, A.	SPECIFICATION AND VERIFICATION TECHNIQUES FOR PARALLEL PROGRAMS BASED ON ...	TR-191
Yun, David Y. Y.	THE HENSEL LEMMA IN ALGEBRAIC MANIPULATION	TR-138

Zachos, S.K.	TWO REMARKS ON THE POWER OF COUNTING	TM-228
Zarmer, C.L.	OFFICE ANALYSIS: METHODOLOGY AND CASE STUDIES	TR-289
Zilles, S.N.	AN EXPANSION OF THE DATA STRUCTURING CAPABILITIES OF PAL	TM-015
Zippel, R.E.	ON THE SECURITY OF THE MERKLE-HELLMAN CRYPTOGRAPHIC SCHEME	TM-119

KEYWORD INDEX

automatic code generation	
AUTOMATIC CODE-GENERATION FROM AN OBJECT-MACHINE ..	TM-018
automatic data-structuring	
SYNTAX-BASED ANALYTIC READING OF MUSICAL SCORES	TR-039
automatic flowcharting	
PROGRAM ANALYSIS BY DIGITAL COMPUTER	TR-043
automatic programming	
AUTOMATIC DESIGN OF DATA PROCESSING SYSTEMS	TM-070
PROTOSYSTEM I· AN AUTOMATIC PROGRAMMING SYSTEM PROTOTYPE	TM-072
INDUCTION IN PROOFS ABOUT PROGRAMS	TR-093
AN ABSTRACT MODEL OF A RESEARCH INSTITUTE: SIMPLE ..	TR-118
A MODEL-DEBUGGING SYSTEM	TR-125
A PROGRAM WRITER	TR-187
DATA DRIVEN LOOPS	TR-244
PRODUCING EXPLANATIONS AND JUSTIFICATIONS OF EXPERT ...	TR-251
SEMIAUTOMATIC TRANSLATION OF COBOL INTO HIBOL	TR-256
axiom system	
LOGICS OF PROGRAMS: AXIOMATICS AND DESCRIPTIVE POWER	TR-200
axiomatic definitions of programming languages	
AXIOMATIC DEFINITIONS OF PROGRAMMING LANGUAGES: A ...	TM-163
AXIOMATIC DEFINITIONS OF PROGRAMMING LANGUAGES, II	TM-179
TERMINATIONS FOR RECURSIVE PROGRAMS: COMPLETENESS AND ..	TM-214
backup	
IMPROVING INFORMATION STORAGE RELIABILITY USING A DATA ...	TM-078
BACKUP AND RECOVERY OF ON-LINE INFORMATION IN A ...	TR-116
Batesons logic	
ON BATESON'S LOGICAL LEVELS OF LEARNING	TM-057
Bayes Rule	
A COMPUTER SYSTEM FOR DECISION ANALYSIS IN HODGKINS ...	TR-271
BCPL	
THE BCPL REFERENCE MANUAL	TR-141
bin packing	
CRITICAL PATH SCHEDULING OF TASK SYSTEMS WITH RESOURCE ...	TM-161
NEAR-OPTIMAL BIN PACKING ALGORITHMS	TR-109
birth defects	
A COMPARATIVE STUDY OF COMPUTER-AIDED CLINICAL DIAGNOSIS	TR-283
bisection size	
ON BISECTING RANDOM GRAPHS	TR-287
bisection	
NEW LOWER BOUND TECHNIQUES FOR VLSI	TM-227
Boolean formula size	
THE COMPLEXITY OF MONOTONE BOOLEAN FUNCTIONS AND AN ...	TR-238
Boolean functions	
A CLASS OF BOOLEAN FUNCTIONS WITH LINEAR COMBINATIONAL ..	TM-055
BICONTINUOUS EXTENSIONS OF INVERTIBLE COMBINATORIAL .	TM-124
Ω (n log n) LOWER BOUNDS ON LENGTH OF BOOLEAN ..	TM-187
THE COMPLEXITY OF MONOTONE BOOLEAN FUNCTIONS AND AN ..	TR-238
branch-and-bound	
AN EFFICIENT ALGORITHM FOR DETERMINING THE LENGTH OF ..	TM-149
building construction	
INTERACTIVE DESIGN COORDINATION FOR THE BUILDING ...	TM-010
building design	
INTERACTIVE DESIGN COORDINATION FOR THE BUILDING .	TM-010
business applications	
A PROGRAM FOR THE DESIGN OF PROCUREMENT SYSTEMS	TR-160
A VERY HIGH LEVEL LANGUAGE FOR BUSINESS DATA PROCESSING	TR-254
C programming language	
A PORTABLE COMPILER FOR THE LANGUAGE C	TR-149
C-element	
CIRCUIT ANALYSIS OF SELF-TIMED ELEMENTS FOR NMOS VLSI ...	TM-220
cable television	
DATA COMMUNICATIONS VIA CABLE TELEVISION NETWORKS: ..	TR-273
calculus problem solving	
CARPS, A PROGRAM WHICH SOLVES CALCULUS WORD PROBLEMS	TR-051
calendar management	
PROGRAMS FOR DISTRIBUTED COMPUTING: THE CALENDAR ...	TM-168
PCAL A PERSONAL CALENDAR	TM-213
COOPERATIVE OFFICE WORK, TELECONFERENCING AND CALENDAR ...	TM-218
cancer	
DIAGNOSTIC PLANNING AND CANCER MANAGEMENT	TR-169
canonic systems	
A CANONIC TRANSLATOR	TR-046
COMPLEXITY MEASURES FOR LANGUAGE RECOGNITION BY CANONIC .	TR-077
canonic translators	
A CANONIC TRANSLATOR	TR-046
canonical form	
TRANSMISSION OF INFORMATION BETWEEN A MAN-MACHINE ..	TM-022
canonical simplification	
ESSAYS IN ALGEBRAIC SIMPLIFICATION	TR-095
capabilities	
TIMESTAMPS AND CAPABILITY-BASED PROTECTION IN A ..	TM-135
capability addressing	
THE ARCHITECTURE OF AN OBJECT BASED PERSONAL COMPUTER	TR-232
cardinality constraints	
ON THE COMPUTATIONAL COMPLEXITY OF CARDINALITY ..	TM-160
Carmichael number	
AN ANALYSIS OF THE SOLOVAY AND STRASSEN TEST FOR ..	TM-108
category alignment	
REFLECTION AND SEMANTICS IN A PROCEDURAL LANGUAGE	TR-272
causal representation	
CAUSAL REPRESENTATION OF PATIENT ILLNESS FOR .	TR-267
cellular arrays	
AUTOMATIC TEST, CONFIGURATION AND REPAIR OF CELLULAR ..	TR-151
cellular automata	
INFORMATION PROCESSING AND TRANSMISSION IN CELLULAR .	TR-081
CGOL	
A FORMALIZATION AND CORRECTNESS PROOF OF THE CGOL .	TR-147
change and contradiction	
SUPPORTING ORGANIZATIONAL PROBLEM SOLVING WITH A ..	TM-224
changing pictures	
THE COMPUTER CONTROL OF CHANGING PICTURES	TR-135
channel routing	
AN APPROXIMATION ALGORITHM FOR MANHATTAN ROUTING	TM-238
ALGORITHMS FOR INTEGRATED CIRCUIT LAYOUT: AN ANALYTIC ...	TR-248
channel width	
WAFER-SCALE INTEGRATION OF SYSTOLIC ARRAYS	TM-236
character generation	
DESIGN OF A LOW-COST CHARACTER GENERATOR FOR REMOTE ...	TR-026
chart parsing	
PRELIMINARY ANALYSIS OF A BREADTH-FIRST PARSING ...	TR-261
Church random sequences	
PSEUDO-RANDOM SEQUENCES	TM-016
circuit and signal models	
MODELS AND DATA STRUCTURES FOR DIGITAL LOGIC SIMULATION	TR-031
circuit complexity	
THE COMPLEXITY OF MONOTONE BOOLEAN FUNCTIONS AND AN ...	TR-238
circuit extraction	
ARTWORK ANALYSIS TOOLS FOR VLSI CIRCUITS	TR-239
circuit layout	
ALGORITHMS FOR INTEGRATED CIRCUIT LAYOUT. AN ANALYTIC ...	TR-248
circuit optimization	
OPTIMIZING SYNCHRONOUS SYSTEMS	TM-215
circuit size	
CIRCUIT-SIZE LOWER BOUNDS AND NON-REDUCIBILITY TO ..	TM-205
classification	
BOUNDS ON INFORMATION RETRIEVAL EFFICIENCY IN STATIC .	TR-088
classroom computer utility model	
THE CLASSROOM INFORMATION AND COMPUTING SERVICE	TR-080
clinical decision making	
DESIGN OF A PROGRAM FOR EXPERT DIAGNOSIS OF ACID BASE ..	TM-132
ARTIFICIAL INTELLIGENCE AND CLINICAL PROBLEM SOLVING	TM-140
A PROGRAM FOR THERAPY OF ACID-BASE AND ELECTROLYTE ...	TM-240
USING INTERACTIVE GRAPHICS IN SIMULATING THE HOSPITAL ..	TR-129
A DIGITALIS THERAPY ADVISOR	TR-143
STRATEGY SELECTION IN MEDICAL DIAGNOSIS	TR-153
DIAGNOSTIC PLANNING AND CANCER MANAGEMENT	TR-169
A DIGITALIS THERAPY ADVISOR WITH EXPLANATIONS	TR-176
CAUSAL REPRESENTATION OF PATIENT ILLNESS FOR . .	TR-267
A COMPUTER SYSTEM FOR DECISION ANALYSIS IN HODGKINS . .	TR-271
A COMPARATIVE STUDY OF COMPUTER-AIDED CLINICAL DIAGNOSIS	TR-283
closures	
REFLECTION AND SEMANTICS IN A PROCEDURAL LANGUAGE	TR-272
clusters	
SEARCH PROCEDURES BASED ON MEASURES OF RELATEDNESS ...	TR-029
CLU	
ABSTRACT DATA TYPES IN STACK BASED LANGUAGES	TR-190
A FORMAL DEFINITION OF CLU	TR-193
A DENOTATIONAL SEMANTICS OF CLU	TR-201
CLU REFERENCE MANUAL	TR-225
TRANSMITTING ABSTRACT VALUES IN MESSAGES	TR-234
COBOL	
SEMIAUTOMATIC TRANSLATION OF COBOL INTO HIBOL	TR-256
code generation	
A PORTABLE COMPILER FOR THE LANGUAGE C	TR-149
code-generation	
AUTOMATIC CODE-GENERATION FROM AN OBJECT-MACHINE .	TM-018
codestripping	
REPRESENTATION AND ANALYSIS OF REAL-TIME CONTROL .	TR-241
codeword sets	
MINIMAX OPTIMAL UNIVERSAL CODEWORD SETS	TM-212
coding techniques	
A FAST SIGNATURE SCHEME	TM-107
collision resolution	
OPTIMAL ARRANGEMENT OF KEYS IN A HASH TABLE	TM-073
colored tokens	
FIRST VERSION OF A DATA FLOW PROCEDURE LANGUAGE	TM-061
combinational complexity	
A CLASS OF BOOLEAN FUNCTIONS WITH LINEAR COMBINATIONAL .	TM-055
THE COMPLEXITY OF NEGATION-LIMITED NETWORKS- A BRIEF .	TM-065

combinatorial analysis	
FINDING ISOMORPH CLASSES FOR COMBINATORIAL STRUCTURES	TM-064
combinatorial optimization	
ON LINEAR CHARACTERIZATIONS OF COMBINATORIAL ...	TM-154
INTEGRAL CONVEX POLYHEDRA AND AN APPROACH TO . .	TR-074
combinatorial problems	
ON REDUCIBILITY AMONG COMBINATORIAL PROBLEMS	TR-113
combinatory algebra	
WHAT IS A MODEL OF THE LAMBDA CALCULUS? EXPANDED ...	TM-201
common base language	
ON THE DESIGN AND SPECIFICATION OF A COMMON BASE ..	TR-101
communicating module	
STREAM-ORIENTED COMPUTATION IN RECURSIVE DATA FLOW ...	TM-068
communication ring	
COMMUNICATION RING INITIALIZATION WITHOUT CENTRAL ...	TM-202
communication	
THE COMPLEXITY OF CONCURRENCY CONTROL FOR DISTRIBUTED ...	TR-269
commutative semigroup	
THE COMPLEXITY OF THE WORD PROBLEMS FOR COMMUTATIVE ...	TM-199
compile generators	
DESIGN AND IMPLEMENTATION OF A TABLE-DRIVEN COMPILER ...	TR-042
compiler meta-languages	
THE SPECIFICATION OF CODE GENERATION ALGORITHMS	TR-199
compiler writing	
PRACTICAL TRANSLATORS FOR LR(K) LANGUAGES	TR-065
THE BCPL REFERENCE MANUAL	TR-141
compilers	
AUTOMATIC CODE-GENERATION FROM AN OBJECT-MACHINE .	TM-018
DESIGN AND IMPLEMENTATION OF A TABLE-DRIVEN COMPILER ...	TR-042
PRACTICAL TRANSLATORS FOR LR(K) LANGUAGES	TR-065
AUTOMATIC CREATION OF A CODE GENERATOR FROM A MACHINE ...	TR-085
A PORTABLE COMPILER FOR THE LANGUAGE C	TR-149
complete axiomatization	
ALGEBRAIC DEPENDENCIES	TM-193
TERMINATIONS FOR RECURSIVE PROGRAMS: COMPLETENESS AND ...	TM-214
THE IMPLICATION PROBLEM FOR FUNCTIONAL AND INCLUSION ...	TM-235
completeness	
A COMPLETENESS RESULT FOR A PROPOSITIONAL DYNAMIC LOGIC	TM-106
DETERMINISTIC PROPOSITIONAL DYNAMIC LOGIC: FINITE ...	TM-190
THE COMPLEXITY OF EVALUATION RELATIONAL QUERIES	TM-229
A SURVEY OF THE LOGIC OF EFFECTIVE DEFINITIONS	TR-246
complex plane diagram	
LAYOUTS FOR THE SHUFFLE-EXCHANGE GRAPH BASED ON THE ...	TM-221
complexity measures	
COMPLEXITY MEASURES FOR PROGRAMMING LANGUAGES	TM-017
COMPLEXITY MEASURES FOR LANGUAGE RECOGNITION BY CANONIC ...	TR-077
complexity	
AN OPERATOR EMBEDDING THEOREM FOR COMPLEXITY CLASSES OF ...	TM-032
FINDING ISOMORPH CLASSES FOR COMBINATORIAL STRUCTURES	TM-064
COMPLEXITY CLASSES OF RECURSIVE FUNCTIONS	TR-110
A CONCEPT OF INDEPENDENCE WITH APPLICATIONS IN VARIOUS ...	TR-235
THE COMPLEXITY OF CONCURRENCY CONTROL FOR DISTRIBUTED ..	TR-269
component placement	
ALGORITHMS FOR INTEGRATED CIRCUIT LAYOUT: AN ANALYTIC ...	TR-248
computability	
THE EQUIVALENCE OF R E. PROGRAMS AND DATA FLOW SCHEMES	TM-121
computation graphs	
THE USE OF QUEUES IN THE PARALLEL DATA FLOW EVALUATION ...	TM-104
computation models	
A GRAPH MODEL FOR PARALLEL COMPUTATIONS	TR-064
computation rate	
ANALYSIS OF ASYNCHRONOUS CONCURRENT SYSTEMS BY TIMED ...	TR-120
computation tree	
LOGICS OF PROGRAMS· AXIOMATICS AND DESCRIPTIVE POWER	TR-200
computation universality	
REVERSIBLE COMPUTING	TM-151
CONSERVATIVE LOGIC	TM-197
computation utility systems	
TRAFFIC CONTROL IN A MULTIPLEXED COMPUTER	TR-030
THE CLASSROOM INFORMATION AND COMPUTING SERVICE	TR-080
computational complexity	
COMPLEXITY MEASURES FOR PROGRAMMING LANGUAGES	TM-017
COMBINING DIMENSIONALITY AND RATE OF GROWTH ARGUMENTS ...	TM-046
THE REDUCTION METHOD FOR ESTABLISHING LOWER BOUNDS ON ...	TM-048
THE INHERENT COMPUTATION COMPLEXITY OF THEORIES OF ...	TM-054
COMPUTATIONAL COMPLEXITY OF THE WORD PROBLEM FOR ...	TM-067
ON THE WORST-CASE OF BEHAVIOR OF STRING-SEARCHING ...	TM-071
LOWER BOUNDS ON INFORMATION TRANSFER IN DISTRIBUTED ...	TM-102
MINIMUM REGISTER ALLOCATION IS COMPLETE IN POLYNOMIAL ...	TM-128
TIME, SPACE AND RANDOMNESS	TM-131
THE SPACE COMPLEXITY OF TWO PEBBLE GAMES ON TREES	TM-133
A SPACE BOUND FOR ONE-TAPE MULTIDIMENSIONAL TURING ...	TM-145
ON THE COMPUTATIONAL COMPLEXITY OF CARDINALITY ...	TM-160
THE COMPLEXITY OF THE WORD PROBLEMS FOR COMMUTATIVE ...	TM-199
COMPUTATIONAL COMPLEXITY AND THE TRAVELING SALESMAN ...	TM-209
RELATIVIZATION OF THE THEORY OF COMPUTATIONAL COMPLEXITY	TR-099
THE COMPLEXITY OF DECISION PROBLEMS IN AUTOMATA THEORY ...	TR-133
NONDETERMINISTIC TIME AND SPACE COMPLEXITY CLASSES	TR-137
SOME UPPER AND LOWER BOUNDS ON DECISION PROCEDURES IN .	TR-139
THE COMPUTATIONAL COMPLEXITY OF SOME LOGICAL THEORIES	TR-144
ON TIME-SPACE CLASSES AND THEIR RELATION TO THE THEORY	TR-195
THE COMPLEXITY OF THE MAXIMUM NETWORK FLOW PROBLEM	TR-230
THE COMPLEXITY OF MONOTONE BOOLEAN FUNCTIONS AND AN ...	TR-238
SIMULATIONS AMONG MULTIDIMENSIONAL TURING MACHINES	TR-242
computational geometry	
A FAST ALGORITHM FOR TESTING FOR SAFETY AND DETECTING ..	TM-181
THE IMPACT OF LAYER ASSIGNMENT METHODS ON LAYOUT . .	TR-291
computational hydrodynamics	
ANALYSIS OF THE SIMPLE CODE FOR DATA FLOW COMPUTATION	TR-216
computational linguistics	
TOWARD A COMPUTATIONAL THEORY OF INDIRECT SPEECH ACTS	TR-223
computational schema	
ASYNCHRONOUS COMPUTATIONAL STRUCTURES	TR-049
COORDINATION OF ASYNCHRONOUS EVENTS	TR-072
computational structures	
ASYNCHRONOUS COMPUTATIONAL STRUCTURES	TR-049
ON THE DESIGN AND SPECIFICATION OF A COMMON BASE .	TR-101
computations in finite fields	
PROBABILISTIC ALGORITHMS IN FINITE FIELDS	TR-213
computer aided design	
CASE STUDY IN INTERACTIVE GRAPHICS PROGRAMMING: A ...	TR-063
computer animation	
INTERACTIVE COMPUTER-MEDIATED ANIMATION	TR-061
computer architecture	
REPORT ON THE WORKSHOP ON DATA FLOW COMPUTER AND ...	TM-092
A LOGIC DESIGN FOR THE CELL BLOCK OF A DATA FLOW . .	TM-093
A COMPUTER ARCHITECTURE FOR DATA FLOW COMPUTATION	TM-100
RESEARCH DIRECTIONS IN COMPUTER ARCHITECTURE	TM-114
REPORT ON THE SECOND WORKSHOP ON DATA FLOW COMPUTER AND ...	TM-136
THE DESIGN OF A MULTIPROCESSOR DEVELOPMENT SYSTEM	TR-279
computer communications	
DESCRIPTION AND FLOW CHART OF THE PDP-7/9 .	TM-011
computer control	
THE DESIGN OF A MODULAR LABORATORY FOR CONTROL ROBOTICS	TM-074
computer culture	
COMPUTERS AND PEOPLE. PERSONAL COMPUTATION	TR-249
computer decentralization	
AN ANALYSIS OF COMPUTER DECENTRALIZATION	TM-090
computer diagnosis	
ARTIFICIAL INTELLIGENCE AND CLINICAL PROBLEM SOLVING	TM-140
computer displays	
DESIGN OF A LOW-COST CHARACTER GENERATOR FOR REMOTE ...	TR-026
computer graphics	
SURFACES FOR COMPUTER-AIDED DESIGN OF SPACE FORMS	TR-041
AN INTEGRATED HARDWARE-SOFTWARE SYSTEM FOR COMPUTER ...	TR-056
INTERACTIVE COMPUTER-MEDIATED ANIMATION	TR-061
EPS: AN INTERACTIVE SYSTEM FOR SOLVING ELLIPTIC ..	TR-062
CASE STUDY IN INTERACTIVE GRAPHICS PROGRAMMING. A ..	TR-063
computer learning	
A COMPUTER MODEL OF SIMPLE FORMS OF LEARNING	TM-020
computer music	
A PARALLEL PROCESSING MODEL OF MUSICAL STRUCTURES	TR-091
computer networks	
MODELING AND DECOMPOSITION OF INFORMATION SYSTEMS FOR . .	TM-026
RESEARCH DIRECTIONS IN COMPUTER ARCHITECTURE	TM-114
PACKET COMMUNICATION	TR-114
FACILITATING INTERPROCESS COMMUNICATION IN A ..	TR-184
DEADLOCK DETECTION IN COMPUTER NETWORKS	TR-185
A MINICOMPUTER NETWORK SIMULATION SYSTEM	TR-208
THE DESIGN OF A ROUTING SERVICE FOR CAMPUS-WIDE ...	TR-270
DATA COMMUNICATIONS VIA CABLE TELEVISION NETWORKS .	TR-273
computer organization	
AN ABSTRACT IMPLEMENTATION FOR A GENERALIZED DATA FLOW ...	TR-228
computer problem solving	
NATURAL LANGUAGE INPUT FOR A COMPUTER PROBLEM SOLVING ..	TR-001
PILOT: A STEP TOWARDS MAN-COMPUTER SYMBIOSIS	TR-032
computer science research	
RESEARCH DIRECTIONS IN COMPUTER ARCHITECTURE	TM-114
computer security	
ENCRYPTION-BASED PROTECTION PROTOCOLS FOR INTERACTIVE ...	TR-162
PROTECTING EXTERNALLY SUPPLIED SOFTWARE IN SMALL ...	TR-255
computer services	
THE COMPUTER UTILITY AS A MARKETPLACE FOR COMPUTER ..	TR-128
computer systems	
MODELING AND DECOMPOSITION OF INFORMATION SYSTEMS FOR ...	TM-026
SIM360: A S/360 SIMULATOR	TM-030
REPORT ON THE SECOND WORKSHOP ON DATA FLOW COMPUTER AND ...	TM-136
COORDINATION OF ASYNCHRONOUS EVENTS	TR-072
PERFORMANCE EVALUATION OF MULTIPROGRAMMED TIME-SHARED ...	TR-103
INTRODUCTION TO MULTICS	TR-123
A MULTI-PROCESS DESIGN OF PAGING SYSTEM	TR-171

COMPUTER SYSTEMS WITH A VERY LARGE ADDRESS SPACE AND ... TR-178
NON-DISCRETIONARY ACCESS CONTROL FOR DECENTRALIZED ... TR-179
SIMULATION OF PACKET COMMUNICATION ARCHITECTURE ... TR-188
FORMAL SPECIFICATIONS FOR PACKET COMMUNICATION SYSTEMS TR-189
INCOMPREHENSIBLE COMPUTER SYSTEMS. KNOWLEDGE WITHOUT ... TR-227
RECOVERY OF THE SWALLOW REPOSITORY TR-252

computer utility
ECONOMIES OF SCALE IN COMPUTER USE. INITIAL TESTS AND ... TR-068
DYNAMIC RECONFIGURATION IN A MODULAR COMPUTER SYSTEM TR-086
COOPERATION OF MUTUALLY SUSPICIOUS SUBSYSTEMS IN A TR-104
MAKING COMPUTERS KEEP SECRETS TR-115
BACKUP AND RECOVERY OF ON-LINE INFORMATION IN A .. TR-116
INTRODUCTION TO MULTICS TR-123
AN EXPERIMENTAL ANALYSIS OF PROGRAM REFERENCE PATTERNS ... TR-127
THE COMPUTER UTILITY AS A MARKETPLACE FOR COMPUTER .. TR-128
REMOVING THE DYNAMIC LINKER FROM THE SECURITY KERNEL OF ... TR-132
MINIMIZING THE NAMING FACILITIES REQUIRING PROTECTION ... TR-156
A SECURE AND FLEXIBLE MODEL OF PROCESS INITIATION FOR A ... TR-163

computer-aided design
VERBAL AND GRAPHICAL LANGUAGE FOR THE AED SYSTEM; A .. TR-004
SURFACES FOR COMPUTER-AIDED DESIGN OF SPACE FORMS TR-041
A SWITCH-LEVEL SIMULATION MODEL FOR INTEGRATED LOGIC .. TR-259

computer-aided diagnosis
A SYSTEM FOR COMPUTER-AIDED DIAGNOSIS TR-044

computer-aided legal analysis
A PRELIMINARY STUDY IN COMPUTER-AIDED LEGAL ANALYSIS TR-157

computer-maintained machines
AUTOMATIC TEST, CONFIGURATION AND REPAIR OF CELLULAR .. TR-151

computers and society
MAKING COMPUTERS KEEP SECRETS TR-115
COMPUTERS AND PEOPLE· PERSONAL COMPUTATION TR-249

concatenation
COMPUTING IN LOGARITHMIC SPACE TM-052

conceptual data models
ON DATA BASE MANAGEMENT SYSTEM ARCHITECTURE TM-141

concurrency control
CONCURRENT AND RELIABLE UPDATES OF DISTRIBUTED DATA BASE TM-144
A THEOREM IN DATA BASE CONCURRENCY CONTROL TM-180
ROBUST CONCURRENCY CONTROL FOR A DISTRIBUTED ... TR-207
THE COMPLEXITY OF CONCURRENCY CONTROL FOR DISTRIBUTED ... TR-269
MULTILEVEL ATOMICITY - A NEW CORRECTNESS CRITERION FOR .. TR-281
CONCURRENCY CONTROL FOR RESILIENT NESTED TRANSACTIONS TR-285
INTERNAL CONSISTENCY OF A DISTRIBUTED TRANSACTION . TR-286

concurrency
CONCURRENT PROGRAMMING TM-115
A SEMANTICS OF SYNCHRONIZATION TM-176
A THEOREM IN DATA BASE CONCURRENCY CONTROL TM-180
A FILE TRANSFER PROGRAM FOR A PERSONAL COMPUTER TM-217
SEMANTICS OF DATA BASE SYSTEMS TR-112
DECIDABILITY QUESTIONS FOR PETRI NETS TR-161
A STRUCTURE MEMORY FOR DATA FLOW COMPUTERS TR-186
SIMULATION OF PACKET COMMUNICATION ARCHITECTURE ... TR-188
ACTORS AND CONTINUOUS FUNCTIONALS TR-194
SYNTHESIS OF SYNCHRONIZATION CODE FOR DATA ABSTRACTIONS TR-203
SYNCHRONIZATION MECHANISMS FOR MODULAR PROGRAMMING . TR-211
THE EVENT BASED LANGUAGE AND ITS MULTIPLE PROCESSOR .. TR-226
AN ABSTRACT IMPLEMENTATION FOR A GENERALIZED DATA FLOW ... TR-228
AUTOMATIC VERIFICATION OF SERIALIZES TR-229

concurrent data base
SERIALIZABILITY OF CONCURRENT DATA BASE UPDATES TR-210

concurrent systems
ANALYSIS OF PRODUCTION SCHEMATA BY PETRI NETS TR-094
ANALYSIS OF ASYNCHRONOUS CONCURRENT SYSTEMS BY TIMED ... TR-120
SCHEDULING TASK SYSTEMS WITH RESOURCES TR-236

confluence
DERIVED PAIRS, OVERLAP CLOSURES, AND REWRITE DOMINOES· . TR-268

consensus
IMPOSSIBILITY OF DISTRIBUTED CONSENSUS WITH ONE FAULTY .. TR-282

conservative logic
CONSERVATIVE LOGIC TM-197

constraint expressions
SOME DATA BASE APPLICATIONS OF CONSTRAINT EXPRESSIONS TR-158

constraints
PHRASE-STRUCTURE PARSING· A METHOD FOR TAKING ADVANTAGE ... TR-296

continuations
ACTOR SYSTEMS FOR REAL-TIME COMPUTATION TR-197
REFLECTION AND SEMANTICS IN A PROCEDURAL LANGUAGE TR-272

continuous extensions
BICONTINUOUS EXTENSIONS OF INVERTIBLE COMBINATORIAL ... TM-124

control flow graph
REPRESENTATION AND ANALYSIS OF REAL-TIME CONTROL ... TR-241

control laboratory
THE DESIGN OF A MODULAR LABORATORY FOR CONTROL ROBOTICS TM-074

control structures
AN ASYNCHRONOUS LOGIC ARRAY TM-062
A MODEL FOR PROCESS REPRESENTATION AND SYNTHESIS TR-087
REAL-TIME CONTROL STRUCTURES FOR BLOCK DIAGRAM SCHEMATA TR-204
REPRESENTATION AND ANALYSIS OF REAL-TIME CONTROL ... TR-241
REFLECTION AND SEMANTICS IN A PROCEDURAL LANGUAGE TR-272

controlled information sharing
CONTROLLED INFORMATION SHARING IN A COMPUTER UTILITY TR-067
PRIMARY ACCESS CONTROL IN LARGE-SCALE TIME-SHARED .. TR-089

conversational maxims
TOWARD A COMPUTATIONAL THEORY OF INDIRECT SPEECH ACTS TR-223

convex polytopes
ON LINEAR CHARACTERIZATIONS OF COMBINATORIAL ... TM-154

cooperating sequential processes
A MODEL FOR PROCESS REPRESENTATION AND SYNTHESIS TR-087

coordinated multiprocessing
COMPUTER DESIGN FOR ASYNCHRONOUSLY REPRODUCIBLE ... TR-034

coordination of events
COORDINATION OF ASYNCHRONOUS EVENTS TR-072

cost analysis
COST ANALYSIS OF DEBUGGING SYSTEMS TR-090

counting problems
TWO REMARKS ON THE POWER OF COUNTING TM-228

Craigs interpolants
A NOTE ON THE LENGTH OF CRAIG'S INTERPOLANTS TM-183

critical pair
DERIVED PAIRS, OVERLAP CLOSURES, AND REWRITE DOMINOES: .. TR-268

crossing number
LAYOUTS FOR THE SHUFFLE-EXCHANGE GRAPH AND LOWER BOUND . TR-274

crossover-free
PLANAR EMBEDDING OF PLANAR GRAPHS TM-237

cryptography
ENCRYPTION SCHEMES FOR COMPUTER CONFIDENTIALITY TM-063
A METHOD FOR OBTAINING SIGNATURES AND PUBLIC-KEY ... TM-082
ON THE SECURITY OF THE MERKLE-HELLMAN CRYPTOGRAPHIC ... TM-119
MENTAL POKER TM-125
ON THE CRYPTOCOMPLEXITY OF KNAPSACK SYSTEMS TM-129
HOW TO SHARE A SECRET TM-134
THE CRYPTOGRAPHIC SECURITY OF COMPACT ... TM-164
EMBEDDING CRYPTOGRAPHIC TRAPDOORS IN ARBITRARY KNAPSACK ... TM-230
RANDOMIZED ENCRYPTION TECHNIQUES TM-234
OCAS - ON-LINE CRYPTANALYTIC AID SYSTEM TR-027
MAKING COMPUTERS KEEP SECRETS TR-115
COMPUTER AND DATA SECURITY: A COMPREHENSIVE ANNOTATED .. TR-122
DIGITALIZED SIGNATURES AND PUBLIC-KEY FUNCTIONS AS .. TR-212
PROTECTING EXTERNALLY SUPPLIED SOFTWARE IN SMALL ... TR-255

cryptology
RANDOMIZED ENCRYPTION TECHNIQUES TM-234
OCAS - ON-LINE CRYPTANALYTIC AID SYSTEM TR-027
MAKING COMPUTERS KEEP SECRETS TR-115
ENCRYPTION-BASED PROTECTION PROTOCOLS FOR INTERACTIVE ... TR-162

CTSS
OPL-I AN OPEN ENDED PROGRAMMING SYSTEM WITHIN CTSS TR-007
THE MAC-SYSTEM A PROGRESS REPORT TR-012
USE OF CTSS IN A TEACHING ENVIRONMENT TR-014
TIME-SHARING ON A MULTICONSOLE COMPUTER TR-017
A SIMULATOR OF MULTIPLE INTERACTIVE USERS TO DRIVE A .. TR-058

cube-connected-cycles graph
AN ASYMPTOTICALLY OPTIMAL LAYOUT FOR THE .. TM-231

cycles
THE SUBGRAPH HOMEOMORPHISM PROBLEM TM-099

cyclic perceptrons
THE COMPLEXITY OF FINITE FUNCTIONS TR-097

daemons
THE COMPUTER CONTROL OF CHANGING PICTURES TR-135

data abstractions
A MANAGER FOR NAMED, PERMANENT OBJECTS TM-162
A FORMALIZATION OF THE STATE MACHINE SPECIFICATION ... TR-202
SYNTHESIS OF SYNCHRONIZATION CODE FOR DATA ABSTRACTIONS TR-203
SYNCHRONIZATION MECHANISMS FOR MODULAR PROGRAMMING ... TR-211
ABSTRACT MODEL SPECIFICATIONS FOR DATA ABSTRACTIONS TR-221
CLU REFERENCE MANUAL TR-225

data analysis
THE OPS-1 MANUAL TR-008

data base concurrency control
A FAST ALGORITHM FOR TESTING FOR SAFETY AND DETECTING TM-181
AN OPTIMALITY THEORY OF CONCURRENCY CONTROL FOR ... TM-185

data base design
A SEMANTIC DATA BASE MODEL AND ITS ASSOCIATED TR-214

data base language
A PRINCIPLED DESIGN FOR AN INTEGRATED COMPUTATIONAL . TM-223

data base management
THE RELATIONAL APPROACH TO THE MANAGEMENT OF DATA BASES TM-023
THE MACAIMS DATA MANAGEMENT SYSTEM TM-024
ON DATA BASE MANAGEMENT SYSTEM ARCHITECTURE TM-141
ATTRIBUTE PARTITIONING IN A SELF-ADAPTIVE RELATIONAL ... TR-192
A SEMANTIC DATA BASE MODEL AND ITS ASSOCIATED .. TR-214

data base modeling
A SEMANTIC DATA BASE MODEL AND ITS ASSOCIATED TR-214
data base operations
SEMANTICS OF DATA BASE SYSTEMS TR-112
data base organization
OPTIMAL ARRANGEMENT OF KEYS IN A HASH TABLE TM-073
data base semantics
ON DATA BASE MANAGEMENT SYSTEM ARCHITECTURE TM-141
data base systems
DATA MODEL EQUIVALENCE TM-118
SEMANTICS OF DATA BASE SYSTEMS TR-112
EQUIVALENCE PROPERTIES OF SEMANTIC DATA MODELS FOR DATA . TR-206
data base transactions
A THEOREM IN DATA BASE CONCURRENCY CONTROL TM-180
data base
ON DATA BASES WITH INCOMPLETE INFORMATION TM-142
GENERALIZED ORGANIZATION OF LARGE DATA BASES A TR-070
SOME DATA BASE APPLICATIONS OF CONSTRAINT EXPRESSIONS TR-158
HIGH LEVEL EXPRESSION OF SEMANTIC INTEGRITY TR-165
INDEX SELECTION IN A SELF-ADAPTIVE RELATIONAL DATA BASE TR-166
MULTILEVEL ATOMICITY - A NEW CORRECTNESS CRITERION FOR TR-281
data definition
A SEMANTIC DATA BASE MODEL AND ITS ASSOCIATED TR-214
data dependencies
ALGEBRAIC DEPENDENCIES TM-193
data dependency programs
THE USE OF QUEUES IN THE PARALLEL DATA FLOW EVALUATION TM-104
data encryption standard
PROTECTING EXTERNALLY SUPPLIED SOFTWARE IN SMALL TR-255
data flow analysis
INTERPROCEDURAL DATA FLOW ANALYSIS IN THE PRESENCE OF TR-247
data flow architecture
DATA-STRUCTURING OPERATIONS IN CONCURRENT COMPUTATIONS TR-224
AN ABSTRACT IMPLEMENTATION FOR A GENERALIZED DATA FLOW TR-228
FAULT TOLERANCE IN PACKET COMMUNICATION COMPUTER TR-250
A DATA FLOW ARCHITECTURE WITH IMPROVED ASYMPTOTIC . TR-265
AN IMPLEMENTATION SCHEME FOR ARRAY OPERATIONS IN STATIC TR-280
data flow computation
THE USE OF QUEUES IN THE PARALLEL DATA FLOW EVALUATION TM-104
BOUNDS ON THE SCHEDULING OF TYPED TASK SYSTEMS TM-111
REPORT ON THE SECOND WORKSHOP ON DATA FLOW COMPUTER AND TM-136
data flow computers
A STRUCTURE MEMORY FOR DATA FLOW COMPUTERS TR-186
data flow computer
A MACHINE LANGUAGE INSTRUCTION SET FOR A DATA FLOW TM-146
data flow instruction cell
A MACHINE LANGUAGE INSTRUCTION SET FOR A DATA FLOW TM-146
data flow languages
A PARALLEL ASYNCHRONOUS COMPUTER ARCHITECTURE FOR DATA TR-150
data flow machine
A PARALLEL ASYNCHRONOUS COMPUTER ARCHITECTURE FOR DATA . TR-150
data flow programming
CONCURRENT PROGRAMMING TM-115
OPERATIONAL SEMANTICS OF A DATA FLOW LANGUAGE TM-120
SAFETY AND OPTIMIZATION TRANSFORMATIONS FOR DATA FLOW TR-240
data flow schema
THE USE OF QUEUES IN THE PARALLEL DATA FLOW EVALUATION TM-104
data flow schemes
THE EQUIVALENCE OF R E PROGRAMS AND DATA FLOW SCHEMES TM-121
data flow translation
SAFETY AND OPTIMIZATION TRANSFORMATIONS FOR DATA FLOW . TR-240
data flow
DECIDABILITY OF EQUIVALENCE FOR A CLASS OF DATA FLOW TM-058
FIRST VERSION OF A DATA FLOW PROCEDURE LANGUAGE TM-061
REPORT ON THE WORKSHOP ON DATA FLOW COMPUTER AND TM-092
A LOGIC DESIGN FOR THE CELL BLOCK OF A DATA FLOW TM-093
A COMPUTER ARCHITECTURE FOR DATA FLOW COMPUTATION TM-100
AN INTERMEDIATE FORM FOR DATA FLOW PROGRAMS TM-143
A DATA FLOW ARCHITECTURE WITH TAGGED TOKENS TM-174
I-STRUCTURES AN EFFICIENT DATA TYPE FOR FUNCTIONAL TM-178
A STRUCTURE MEMORY FOR DATA FLOW COMPUTERS TR-186
ANALYSIS OF THE SIMPLE CODE FOR DATA FLOW COMPUTATION TR-216
DENOTATIONAL SEMANTICS OF DETERMINATE AND TR-220
THE EVENT BASED LANGUAGE AND ITS MULTIPLE PROCESSOR TR-226
APPLICATION OF DATA FLOW ARCHITECTURE TO COMPUTER MUSIC . TR-257
data links
DESCRIPTION AND FLOW CHART OF THE PDP-7/9 TM-011
data management system
THE RELATIONAL APPROACH TO THE MANAGEMENT OF DATA BASES TM-023
THE MACAIMS DATA MANAGEMENT SYSTEM TM-024
data management
GENERALIZED ORGANIZATION OF LARGE DATA BASES A TR-070
DESIGN STRATEGIES FOR FILE SYSTEMS TR-078
data manipulation
THE RELATIONAL APPROACH TO THE MANAGEMENT OF DATA BASES TM-023
THE MACAIMS DATA MANAGEMENT SYSTEM TM-024
data model equivalence
DATA MODEL EQUIVALENCE TM-118
EQUIVALENCE PROPERTIES OF SEMANTIC DATA MODELS FOR DATA TR-206
data models
EQUIVALENCE PROPERTIES OF SEMANTIC DATA MODELS FOR DATA TR-206
A SEMANTIC DATA BASE MODEL AND ITS ASSOCIATED . TR-214
data organization
THE RELATIONAL APPROACH TO THE MANAGEMENT OF DATA BASES TM-023
THE MACAIMS DATA MANAGEMENT SYSTEM TM-024
data security
COMPUTER AND DATA SECURITY A COMPREHENSIVE ANNOTATED TR-122
data semantics
A SEMANTIC DATA BASE MODEL AND ITS ASSOCIATED . TR-214
data storage
IMPROVED BOUNDS ON THE COSTS OF OPTIMAL AND BALANCED . TM-069
data streams
STREAM-ORIENTED COMPUTATION IN RECURSIVE DATA FLOW TM-068
data structures
AN EXPANSION OF THE DATA STRUCTURING CAPABILITIES OF PAL TM-015
AN INVESTIGATION OF CURRENT LANGUAGE SUPPORT FOR THE TM-051
FIRST VERSION OF A DATA FLOW PROCEDURE LANGUAGE TM-061
I-STRUCTURES AN EFFICIENT DATA TYPE FOR FUNCTIONAL TM-178
VERIFICATION OF PROGRAMS OPERATING ON STRUCTURED DATA TR-124
SEMANTICS OF DATA STRUCTURES AND REFERENCES TR-134
data structure
THE RELATIONAL APPROACH TO THE MANAGEMENT OF DATA BASES TM-023
THE MACAIMS DATA MANAGEMENT SYSTEM TM-024
GENERALIZED ORGANIZATION OF A LARGE DATA BASES A TR-070
data types
I-STRUCTURES AN EFFICIENT DATA TYPE FOR FUNCTIONAL TM-178
ABSTRACT MODEL SPECIFICATIONS FOR DATA ABSTRACTIONS TR-221
TOWARDS A THEORY FOR ABSTRACT DATA TYPES TR-237
deadline-driven scheduling
REAL-TIME CONTROL STRUCTURES FOR BLOCK DIAGRAM SCHEMATA TR-204
deadlock detection
DEADLOCK DETECTION IN COMPUTER NETWORKS TR-185
deadlock
DEADLOCK-FREE SHARING OF RESOURCES IN ASYNCHRONOUS TR-075
SYNTHESIS OF SYNCHRONIZATION CODE FOR DATA ABSTRACTIONS TR-203
debugging
A DYNAMIC DEBUGGING SYSTEM FOR MDL TM-094
COST ANALYSIS OF DEBUGGING SYSTEMS TR-090
A MODEL-DEBUGGING SYSTEM TR-125
A ROBUST ENVIRONMENT FOR PROGRAM DEVELOPMENT TR-175
INTERACTIVE DEBUGGING IN A DISTRIBUTED COMPUTATIONAL TR-264
REFLECTION AND SEMANTICS IN A PROCEDURAL LANGUAGE TR-272
decentralized computing systems
NON-DISCRETIONARY ACCESS CONTROL FOR DECENTRALIZED TR-179
decidability of equivalence
DECIDABILITY OF EQUIVALENCE FOR A CLASS OF DATA FLOW TM-058
decidability
A DECIDABILITY RESULT FOR A SECOND ORDER PROCESS LOGIC TM-12
decision analysis
A COMPUTER SYSTEM FOR DECISION ANALYSIS IN HODGKINS TR-271
decision method
A NEAR-OPTIMAL METHOD FOR REASONING ABOUT ACTION TM-113
PROPOSITIONAL DYNAMIC LOGIC OF LOOPING AND CONVERSE TR-263
decision problems
DECISION PROBLEMS FOR PETRINETS AND VECTOR ADDITION TM-059
FORMAL PROPERTIES OF WELL-FORMED DATA FLOW SCHEMAS TM-066
COMPUTATIONAL COMPLEXITY OF THE WORD PROBLEM FOR TM-067
decision procedure
THE PROPOSITIONAL DYNAMIC LOGIC OF DETERMINISTIC, TM-198
THE COMPLEXITY OF DECISION PROBLEMS IN AUTOMATA THEORY TR-133
decision-support system
A COMPUTER SYSTEM FOR DECISION ANALYSIS IN HODGKINS TR-271
DECNET
A NETWORK TRAFFIC GENERATOR FOR DECNET TM-127
decomposition
MODELING AND DECOMPOSITION OF INFORMATION SYSTEMS FOR TM-026
denotational semantics
A DENOTATIONAL SEMANTICS OF CLU TR-201
DENOTATIONAL SEMANTICS OF DETERMINATE AND TR-220
depth cues
SHAPE FROM SHADING A METHOD FOR OBTAINING THE SHAPE OF TR-079
derived pair
DERIVED PAIRS, OVERLAP CLOSURES, AND REWRITE DOMINOES TR-268
descriptive complexity
PSEUDO-RANDOM SEQUENCES TM-016
design automation
THE IMPACT OF LAYER ASSIGNMENT METHODS ON LAYOUT TR-291
FUNDAMENTAL DESIGN PROBLEMS OF DISTRIBUTED SYSTEMS FOR TR-297
design methodology
OPTIMIZING SYNCHRONOUS SYSTEMS TM-215

design of simulation experiments	
IN-PROCESS MANUFACTURING QUALITY CONTROL	TR-083
design rule checking	
ARTWORK ANALYSIS TOOLS FOR VLSI CIRCUITS	TR-239
desk-to-desk conferencing	
SOFTWARE FOR THE 'ROLES' PEOPLE PLAY	TM-210
COOPERATIVE OFFICE WORK, TELECONFERENCING AND CALENDAR	TM-218
determinacy	
THE FLOW GRAPH SCHEMATA MODEL OF PARALLEL COMPUTATION	TR-053
DATA-STRUCTURING OPERATIONS IN CONCURRENT COMPUTATIONS	TR-224
determinism	
ON MEMORY LIMITATIONS IN NATURAL LANGUAGE PROCESSING	TR-245
deterministic dynamic logic	
ON THE EXPRESSIVE POWER OF DYNAMIC LOGIC, II	TM-204
deterministic propositional dynamic logic	
DETERMINISTIC PROPOSITIONAL DYNAMIC LOGIC FINITE	TM-190
diagnostics	
DIAGNOSTIC PLANNING AND CANCER MANAGEMENT	TR-169
dialogue	
A SYSTEM TO PROCESS DIALOGUE· A PROGRESS REPORT	TM-079
differential geometry	
COMPUTER PROGRAMS FOR RESEARCH IN GRAVITATION AND	TM-167
digital computation	
ON THE SIMULATION OF DYNAMIC SYSTEMS WITH LUMPED	TR-045
digital logic simulation	
MODELS AND DATA STRUCTURES FOR DIGITAL LOGIC SIMULATION	TR-031
digital logic	
COORDINATION OF ASYNCHRONOUS EVENTS	TR-072
digital signatures	
A METHOD FOR OBTAINING SIGNATURES AND PUBLIC-KEY	TM-082
ON THE CRYPTOCOMPLEXITY OF KNAPSACK SYSTEMS	TM-129
digital simulation	
ON THE SIMULATION OF DYNAMIC SYSTEMS WITH LUMPED	TR-045
digital systems	
AN ASYNCHRONOUS LOGIC ARRAY	TM-062
THE DESIGN OF A MODULAR LABORATORY FOR CONTROL ROBOTICS	TM-074
REPORT ON THE WORKSHOP ON SELF-TIMED SYSTEMS	TM-166
digital/analog systems	
A LOW-COST OUTPUT TERMINAL FOR TIME-SHARED COMPUTERS	TR-038
digitalis	
A DIGITALIS THERAPY ADVISOR	TR-143
A DIGITALIS THERAPY ADVISOR WITH EXPLANATIONS	TR-176
digitalized signatures	
DIGITALIZED SIGNATURES AND PUBLIC-KEY FUNCTIONS AS	TR-212
directed graphs	
MINIMUM REGISTER ALLOCATION IS COMPLETE IN POLYNOMIAL	TM-128
THE COMPLEXITY OF MONOTONE BOOLEAN FUNCTIONS AND AN .	TR-238
directories	
BOUNDS ON INFORMATION RETRIEVAL EFFICIENCY IN STATIC	TR-088
discrete computation	
DISCRETE COMPUTATION THEORY AND OPEN PROBLEMS	TM-039
discrete mechanics	
CONSERVATIVE LOGIC	TM-197
display management	
DISPLAY MANAGEMENT IN AN INTEGRATED OFFICE	TR-278
display processor	
A COMPUTER-CONTROLLED GRAPHICAL DISPLAY PROCESSOR	TR-071
display software systems	
AN INTEGRATED HARDWARE-SOFTWARE SYSTEM FOR COMPUTER .	TR-056
CASE STUDY IN INTERACTIVE GRAPHICS PROGRAMMING A	TR-063
distributed computer systems	
RESEARCH DIRECTIONS IN COMPUTER ARCHITECTURE	TM-114
TIMESTAMPS AND CAPABILITY-BASED PROTECTION IN A	TM-135
ACTORS AND CONTINUOUS FUNCTIONALS	TR-194
NAMING AND SYNCHRONIZATION IN A DECENTRALIZED COMPUTER	TR-205
DISTRIBUTED COMPUTER SYSTEMS: STRUCTURE AND SEMANTICS	TR-215
distributed computing	
PROGRAMS FOR DISTRIBUTED COMPUTING THE CALENDAR	TM-168
MULTIPLE-PROCESSOR IMPLEMENTATIONS OF MESSAGE PASSING	TR-198
REFERENCE TREE NETWORKS· VIRTUAL MACHINE AND	TR-222
NESTED TRANSACTIONS AN APPROACH TO RELIABLE	TR-260
IMPOSSIBILITY OF DISTRIBUTED CONSENSUS WITH ONE FAULTY	TR-282
distributed data base	
CONCURRENT AND RELIABLE UPDATES OF DISTRIBUTED DATA BASE	TM-144
TIMEPAD - A PERFORMANCE IMPROVING SYNCHRONIZATION	TM-177
THE COMPLEXITY OF CONCURRENCY CONTROL FOR DISTRIBUTED	TR-269
IMPOSSIBILITY OF DISTRIBUTED CONSENSUS WITH ONE FAULTY	TR-282
distributed data storage system	
RECOVERY OF THE SWALLOW REPOSITORY	TR-252
distributed object management	
REFERENCE TREE NETWORKS VIRTUAL MACHINE AND	TR-222
distributed processing	
LOWER BOUNDS ON INFORMATION TRANSFER IN DISTRIBUTED	TM-102
distributed systems	
IMPROVING INFORMATION STORAGE RELIABILITY USING A DATA	TM-078
ANALYSIS OF ASYNCHRONOUS CONCURRENT SYSTEMS BY TIMED .	TR-120
ROBUST CONCURRENCY CONTROL FOR A DISTRIBUTED	TR-207
COPYING COMPLEX STRUCTURES IN A DISTRIBUTED SYSTEM	TR-219
TRANSMITTING ABSTRACT VALUES IN MESSAGES	TR-234
MANAGEMENT OF OBJECT HISTORIES IN THE SWALLOW REPOSITORY	TR-243
PROTECTING EXTERNALLY SUPPLIED SOFTWARE IN SMALL	TR-255
INTERACTIVE DEBUGGING IN A DISTRIBUTED COMPUTATIONAL	TR-264
FUNDAMENTAL DESIGN PROBLEMS OF DISTRIBUTED SYSTEMS FOR	TR-297
divergence	
NONDETERMINISM IN LOGICS OF PROGRAMS	TM-098
ARITHMETICAL COMPLETENESS IN LOGICS OF PROGRAMS	TM-103
LOGICS OF PROGRAMS AXIOMATICS AND DESCRIPTIVE POWER	TR-200
document processing	
AN INTEGRATED APPROACH TO FORMATTED DOCUMENT PRODUCTION	TR-253
document searching	
SEARCH PROCEDURES BASED ON MEASURES OF RELATEDNESS	TR-029
dual-rail code	
CIRCUIT ANALYSIS OF SELF-TIMED ELEMENTS FOR NMOS VLSI	TM-220
dynamic algebra	
DYNAMIC ALGEBRAS EXAMPLES, CONSTRUCTIONS, APPLICATIONS	TM-138
DYNAMIC ALGEBRAS AND THE NATURE OF INDUCTION	TM-159
dynamic linker	
REMOVING THE DYNAMIC LINKER FROM THE SECURITY KERNEL OF	TR-132
dynamic logic	
COMPUTABILITY AND COMPLETENESS IN LOGICS OF PROGRAMS	TM-097
NONDETERMINISM IN LOGICS OF PROGRAMS	TM-098
ARITHMETICAL COMPLETENESS IN LOGICS OF PROGRAMS	TM-103
A COMPLETENESS RESULT FOR A PROPOSITIONAL DYNAMIC LOGIC	TM-106
A DECIDABILITY RESULT FOR A SECOND ORDER PROCESS LOGIC	TM-112
A NEAR-OPTIMAL METHOD FOR REASONING ABOUT ACTION	TM-113
APPLICATIONS OF MODAL LOGIC TO PROGRAMMING	TM-116
SIX LECTURES ON DYNAMIC LOGIC	TM-117
DEFINABILITY IN DYNAMIC LOGIC	TM-156
ON THE EXPRESSIVE POWER OF DYNAMIC LOGIC	TM-157
THE DEDUCIBILITY PROBLEM IN PROPOSITIONAL DYNAMIC LOGIC	TM-192
ON THE EXPRESSIVE POWER OF DYNAMIC LOGIC, II	TM-204
A NOTE ON EQUIVALENCES AMONG LOGICS OF PROGRAMS	TM-211
LOGICS OF PROGRAMS: AXIOMATICS AND DESCRIPTIVE POWER	TR-200
PROPOSITIONAL DYNAMIC LOGIC OF LOOPING AND CONVERSE	TR-263
dynamic monitoring	
THE GRAPHIC DISPLAY AS AN AID IN THE MONITORING OF A	TR-054
dynamic programming	
THE TRAVELING SALESMAN PROBLEM WITH MANY VISITS TO FEW	TM-208
dynamic reconfiguration	
DYNAMIC RECONFIGURATION IN A MODULAR COMPUTER SYSTEM	TR-086
dynamic redundancy	
FAULT TOLERANCE IN PACKET COMMUNICATION COMPUTER	TR-250
dynamic scoping	
REFLECTION AND SEMANTICS IN A PROCEDURAL LANGUAGE	TR-272
dynamic systems	
ON THE SIMULATION OF DYNAMIC SYSTEMS WITH LUMPED	TR-045
Early's algorithm	
PRELIMINARY ANALYSIS OF A BREADTH-FIRST PARSING	TR-261
EBL	
THE EVENT BASED LANGUAGE AND ITS MULTIPLE PROCESSOR	TR-226
economics of manufacturing	
IN-PROCESS MANUFACTURING QUALITY CONTROL	TR-083
economics	
ECONOMIES OF SCALE IN COMPUTER USE· INITIAL TESTS AND	TR-068
edge connectivity	
COVERING GRAPHS BY SIMPLE CIRCUITS	TM-155
Edmonds-Karp scaling method	
THE TRAVELING SALESMAN PROBLEM WITH MANY VISITS TO FEW	TM-208
education	
USE OF CTSS IN A TEACHING ENVIRONMENT	TR-014
effective definitions	
A SURVEY OF THE LOGIC OF EFFECTIVE DEFINITIONS	TR-246
effective functionals	
THE EQUIVALENCE OF R E PROGRAMS AND DATA FLOW SCHEMES	TM-121
effectiveness	
EFFECTIVENESS	TM-109
electronic funds transfer	
A METHOD FOR OBTAINING SIGNATURES AND PUBLIC-KEY	TM-082
electronic mail	
A METHOD FOR OBTAINING SIGNATURES AND PUBLIC-KEY	TM-082
electronic signatures	
A FAST SIGNATURE SCHEME	TM-107
elliptical-boundary value problems	
EPS AN INTERACTIVE SYSTEM FOR SOLVING ELLIPTIC	TR-062
embedded implicational dependencies	
ALGEBRAIC DEPENDENCIES	TM-193
embedded systems	
FUNDAMENTAL DESIGN PROBLEMS OF DISTRIBUTED SYSTEMS FOR	TR-297
emptiness problem	
THE EMPTINESS PROBLEM FOR AUTOMATA ON INFINITE TREES	TM-029

functional modularity
TRANSMISSION OF INFORMATION BETWEEN A MAN-MACHINE .. TM-022
functional programming
REFLECTION AND SEMANTICS IN A PROCEDURAL LANGUAGE TR-272
AN IMPLEMENTATION SCHEME FOR ARRAY OPERATIONS IN STATIC TR-280
functional semantics
I-STRUCTURES AN EFFICIENT DATA TYPE FOR FUNCTIONAL . TM-178
Galois groups
ON COMPUTING GALOIS GROUPS AND ITS APPLICATION TO TR-288
garbage collection
A NEW LIST-TRACING ALGORITHM TM-019
A REAL TIME GARBAGE COLLECTOR THAT CAN RECOVER . TM-184
COMPUTER SYSTEMS WITH A VERY LARGE ADDRESS SPACE AND TR-178
ABSTRACT DATA TYPES IN STACK BASED LANGUAGES TR-190
ACTOR SYSTEMS FOR REAL-TIME COMPUTATION TR-197
A MACHINE ARCHITECTURE TO SUPPORT AN OBJECT-ORIENTED TR-209
general canonic systems
FURTHER RESULTS ON HIERARCHIES OF CANONIC SYSTEMS TR-100
general relativity
COMPUTER PROGRAMS FOR RESEARCH IN GRAVITATION AND . TM-167
geometric complexity
THE COMPLEXITY OF THE MAXIMUM NETWORK FLOW PROBLEM TR-230
geometry of locking
A FAST ALGORITHM FOR TESTING FOR SAFETY AND DETECTING . TM-181
geometry
CONSTRUCTION HEURISTICS FOR GEOMETRY AND A VECTOR TM-028
Graham-Shamir knapsacks
ON THE SECURITY OF THE MERKLE-HELLMAN CRYPTOGRAPHIC TM-119
grammars
PHRASE-STRUCTURE PARSING: A METHOD FOR TAKING ADVANTAGE TR-296
grammar
EFFECTIVENESS TM-109
grammatical transformation
A NEW GRAMMATICAL TRANSFORMATION INTO DETERMINISTIC TR-119
graph algorithms
COVERING GRAPHS BY SIMPLE CIRCUITS TM-155
graph embedding
LAYOUTS FOR THE SHUFFLE-EXCHANGE GRAPH BASED ON THE TM-221
NEW LOWER BOUND TECHNIQUES FOR VLSI TM-227
AN ASYMPTOTICALLY OPTIMAL LAYOUT FOR THE . TM-231
PLANAR EMBEDDING OF PLANAR GRAPHS TM-237
graph representation
AN INTERMEDIATE FORM FOR DATA FLOW PROGRAMS TM-143
graph separator
LAYOUTS FOR THE SHUFFLE-EXCHANGE GRAPH AND LOWER BOUND TR-274
graph theory
OPTIMAL PLACEMENT FOR RIVER ROUTING TM-206
OPTIMIZING SYNCHRONOUS SYSTEMS TM-215
THE IMPACT OF LAYER ASSIGNMENT METHODS ON LAYOUT TR-291
graphic data display
THE GRAPHIC DISPLAY AS AN AID IN THE MONITORING OF A . TR-054
graphic display
SYMBOLIC MATHEMATICAL LABORATORY TR-036
A COMPUTER-CONTROLLED GRAPHICAL DISPLAY PROCESSOR TR-071
graphical input-output
MAP· A SYSTEM FOR ON-LINE MATHEMATICAL ANALYSIS TR-024
graphics
A DYNAMIC DEBUGGING SYSTEM FOR MDL TM-094
USING INTERACTIVE GRAPHICS IN SIMULATING THE HOSPITAL TR-129
gravitation
COMPUTER PROGRAMS FOR RESEARCH IN GRAVITATION AND . TM-167
grid graphs
HAMILTON PATHS IN GRID GRAPHS TM-182
group manipulation programs
CAMAC GROUP MANIPULATION SYSTEM TM-060
group theory
ADEPT - A HEURISTIC PROGRAM FOR PROVING THEOREMS OF TR-033
guarded commands
LOGICS OF PROGRAMS· AXIOMATICS AND DESCRIPTIVE POWER TR-200
Hamilton circuit
HAMILTON PATHS IN GRID GRAPHS TM-182
COMPUTATIONAL COMPLEXITY AND THE TRAVELING SALESMAN TM-209
ard real-time systems
FUNDAMENTAL DESIGN PROBLEMS OF DISTRIBUTED SYSTEMS FOR TR-297
ardness
THE COMPLEXITY OF EVALUATION RELATIONAL QUERIES TM-229
ardware synthesizers
APPLICATION OF DATA FLOW ARCHITECTURE TO COMPUTER MUSIC TR-257
ash table
RECOVERY OF THE SWALLOW REPOSITORY TR-252
ashing
OPTIMAL ARRANGEMENT OF KEYS IN A HASH TABLE TM-073
eap
FIRST VERSION OF A DATA FLOW PROCEDURE LANGUAGE TM-061

Henkin prefix
CHARACTERIZING SECOND ORDER LOGIC WITH FIRST ORDER . TM-095
heterogeneous
FURTHER RESULTS ON HIERARCHIES OF CANONIC SYSTEMS TR-100
heuristic algorithms
ALGORITHMS FOR INTEGRATED CIRCUIT LAYOUT: AN ANALYTIC . TR-248
heuristic programming
PILOT· A STEP TOWARDS MAN-COMPUTER SYMBIOSIS TR-032
heuristics
CONSTRUCTION HEURISTICS FOR GEOMETRY AND A VECTOR TM-028
COMPUTER RECOGNITION OF THREE-DIMENSIONAL OBJECTS IN A TR-059
HIBOL
SEMIAUTOMATIC TRANSLATION OF COBOL INTO HIBOL TR-256
hierarchy
FURTHER RESULTS ON HIERARCHIES OF CANONIC SYSTEMS TR-100
STORAGE HIERARCHY SYSTEMS TR-107
high level languages
USE OF HIGH LEVEL LANGUAGES FOR SYSTEMS PROGRAMMING TM-013
AUTOMATIC DESIGN OF DATA PROCESSING SYSTEMS TM-070
PROTOSYSTEM I· AN AUTOMATIC PROGRAMMING SYSTEM PROTOTYPE TM-072
high level machine architecture
THE ARCHITECTURE OF AN OBJECT BASED PERSONAL COMPUTER TR-232
high-performance computers
A DATA FLOW ARCHITECTURE WITH TAGGED TOKENS TM-174
higher order
REFLECTION AND SEMANTICS IN A PROCEDURAL LANGUAGE TR-272
Hintikka set
A NEAR-OPTIMAL METHOD FOR REASONING ABOUT ACTION TM-113
Hoare logic
A COMPLETE AXIOMATIC SYSTEM FOR PROVING DEDUCTIONS TM-096
Hoare Logic
HOARES'S LOGIC IS NOT COMPLETE WHEN IT COULD BE TM-226
Hodgkins disease
DIAGNOSTIC PLANNING AND CANCER MANAGEMENT TR-169
A COMPUTER SYSTEM FOR DECISION ANALYSIS IN HODGKINS TR-271
homeomorphisms of graphs
THE SUBGRAPH HOMEOMORPHISM PROBLEM TM-099
human factors
AN EASE OF USE EVALUATION OF AN INTEGRATED EDITOR AND TR-266
ICES system
INTERACTIVE DESIGN COORDINATION FOR THE BUILDING TM-010
if-then-while programs
THE USE OF QUEUES IN THE PARALLEL DATA FLOW EVALUATION TM-104
image processing
SHAPE FROM SHADING A METHOD FOR OBTAINING THE SHAPE OF TR-079
implementation languages
A PORTABLE COMPILER FOR THE LANGUAGE C TR-149
implicit information
ON DATA BASES WITH INCOMPLETE INFORMATION TM-142
inclusion dependency
THE IMPLICATION PROBLEM FOR FUNCTIONAL AND INCLUSION TM-235
incomplete information
ON DATA BASES WITH INCOMPLETE INFORMATION TM-142
incomprehensible systems
INCOMPREHENSIBLE COMPUTER SYSTEMS KNOWLEDGE WITHOUT . TR-227
incremental simulation
INCREMENTAL SIMULATION ON A TIME-SHARED COMPUTER TR-048
incrementally non-blocking connectors
ON CONCENTRATION AND CONNECTION NETWORKS TM-196
independence
A CONCEPT OF INDEPENDENCE WITH APPLICATIONS IN VARIOUS . TR-235
indirect speech acts
TOWARD A COMPUTATIONAL THEORY OF INDIRECT SPEECH ACTS TR-223
industry installations
ECONOMIES OF SCALE IN COMPUTER USE INITIAL TESTS AND TR-068
infinitary logic
DEFINABILITY IN DYNAMIC LOGIC TM-156
A SURVEY OF THE LOGIC OF EFFECTIVE DEFINITIONS TR-246
infinite loops
PROPOSITIONAL DYNAMIC LOGIC OF LOOPING AND CONVERSE TR-263
infinite tree automata
THE EMPTINESS PROBLEM FOR AUTOMATA ON INFINITE TREES TM-029
THE EMPTINESS AND COMPLEMENTATION PROBLEMS FOR AUTOMATA . TR-106
information flow
THE LOGIC OF SYSTEMS TR-170
information mechanics
CONSERVATIVE LOGIC TM-197
information retrieval
THE RELATIONAL APPROACH TO THE MANAGEMENT OF DATA BASES TM-023
THE MACAIMS DATA MANAGEMENT SYSTEM TM-024
MDC-PROGRAMMER A MUDDLE-TO-DATALANGUAGE TRANSLATOR FOR TM-053
SEARCH PROCEDURES BASED ON MEASURES OF RELATEDNESS . TR-029
BOUNDS ON INFORMATION RETRIEVAL EFFICIENCY IN STATIC TR-088
information sharing
CONTROLLED INFORMATION SHARING IN A COMPUTER UTILITY TR-067

information systems
TRANSMISSION OF INFORMATION BETWEEN A MAN-MACHINE TM-022
MODELING AND DECOMPOSITION OF INFORMATION SYSTEMS FOR TM-026

information theory
TIME, SPACE AND RANDOMNESS TM-131
BOUNDS ON INFORMATION RETRIEVAL EFFICIENCY IN STATIC TR-088

input/output architecture
AN INPUT/OUTPUT ARCHITECTURE FOR VIRTUAL MEMORY . TR-117

input/output
INPUT/OUTPUT IN TIME-SHARED, SEGMENTED, MULTIPROCESSOR TR-028

instruction cells
A MACHINE LANGUAGE INSTRUCTION SET FOR A DATA FLOW . TM-146
HIGH LEVEL VAL CONSTRUCTS IN A STATIC DATA FLOW MACHINE TR-262

instruction set
A MACHINE LANGUAGE INSTRUCTION SET FOR A DATA FLOW TM-146

integer linear programming
ON THE COMPLEXITY OF INTEGER PROGRAMMING TM-152
INTEGRAL CONVEX POLYHEDRA AND AN APPROACH TO TR-074

integer multiplication
FAST ON-LINE INTEGER MULTIPLICATION TM-045

integral convex hulls
INTEGRAL CONVEX POLYHEDRA AND AN APPROACH TO TR-074

integral convex polyhedra
INTEGRAL CONVEX POLYHEDRA AND AN APPROACH TO TR-074

integrated computational environment
A PRINCIPLED DESIGN FOR AN INTEGRATED COMPUTATIONAL TM-223

integrated office systems
ANALYSIS AND SPECIFICATION OF OFFICE PROCEDURES TR-275

intention
REFLECTION AND SEMANTICS IN A PROCEDURAL LANGUAGE TR-272

interactive computing
ABSENTEE COMPUTATIONS IN A MULTIPLE-ACCESS COMPUTER TR-052
COST ANALYSIS OF DEBUGGING SYSTEMS TR-090

interactive graphics
INTERACTIVE COMPUTER-MEDIATED ANIMATION TR-061

interactive problem solving
EPS AN INTERACTIVE SYSTEM FOR SOLVING ELLIPTIC TR-062

interactive systems
THE SUBSTANTIVE USE OF COMPUTERS FOR INTELLECTUAL TM-021
HELPING PEOPLE THINK TM-025

intermodule dependencies
A CASE STUDY OF INTERMODULE DEPENDENCIES IN A VIRTUAL TR-174

internal consistency
INTERNAL CONSISTENCY OF A DISTRIBUTED TRANSACTIÒN . TR-286

internet remote login
IMPLEMENTING INTERNET REMOTE LOGIN ON A PERSONAL . TM-233

interpolation
HOW TO SHARE A SECRET TM-134

interpretation
REFLECTION AND SEMANTICS IN A PROCEDURAL LANGUAGE TR-272

interpreters
COST ANALYSIS OF DEBUGGING SYSTEMS TR-090

interprocess commnication
FACILITATING INTERPROCESS COMMUNICATION IN A TR-184
ACTORS AND CONTINUOUS FUNCTIONALS TR-194
SYNTHESIS OF SYNCHRONIZATION CODE FOR DATA ABSTRACTIONS TR-203

interrupts
SUSPENSION OF PROCESSES IN A MULTIPROCESSING COMPUTER TM-014
A MODEL FOR PROCESS REPRESENTATION AND SYNTHESIS TR-087
REPRESENTATION AND ANALYSIS OF REAL-TIME CONTROL . TR-241

intractable problems
DIGITALIZED SIGNATURES AND PUBLIC-KEY FUNCTIONS AS TR-212

invariant subcodes
SYMMETRY CODES AND THEIR INVARIANT SUBCODES TM-044

invertible combinatorial functions
BICONTINUOUS EXTENSIONS OF INVERTIBLE COMBINATORIAL TM-124

isomorph classes
FINDING ISOMORPH CLASSES FOR COMBINATORIAL STRUCTURES TM-064

iteractive arrays
RECOGNITION OF TOPOLOGICAL INVARIANTS BY ITERACTIVE TR-066

iteration abstractions
CLU REFERENCE MANUAL TR-225

iteration theorems
PUMPING LEMMAS FOR REGULAR SETS TM-170

iteration
LOOP ITERATION MACRO TM-169

iterative arrays
INFORMATION PROCESSING AND TRANSMISSION IN CELLULAR TR-081

J-K flip-flops
MATHEMATICAL FOUNDATIONS OF FLIP-FLOPS TM-047
ENCRYPTION SCHEMES FOR COMPUTER CONFIDENTIALITY TM-063
AUTONOMOUS, SYNCHRONOUS COUNTERS CONSTRUCTED ONLY OF . TR-096

Jacobi symbol
AN ANALYSIS OF THE SOLOVAY AND STRASSEN TEST FOR . TM-108

jurismetrics
A PRELIMINARY STUDY IN COMPUTER-AIDED LEGAL ANALYSIS TR-157

K-median problem
WORST-CASE AND PROBABILISTIC ANALYSIS OF A GEOMETRIC TM-153

Karzanov algorithm
THE MAX FLOW ALGORITHM OF DINIC AND KARZANOV AN TM-080
CONSTRUCTION AND ANALYSIS OF NETWORK FLOW PROBLEM WHICH TM-083

kernel-structured operating system
PROCESSOR MULTIPLEXING IN A LAYERED OPERATING SYSTEM TR-164

key management
HOW TO SHARE A SECRET TM-134

Khacians algorithm
ON LINEAR CHARACTERIZATIONS OF COMBINATORIAL TM-154

Kleene hierarchy
PSEUDO-RANDOM SEQUENCES TM-016

knapsack problems
A FAST SIGNATURE SCHEME TM-107
ON THE CRYPTOCOMPLEXITY OF KNAPSACK SYSTEMS TM-129
A $T = 0(2^{n/2})$, $S = 0(2^{n/4})$ ALGORITHM FOR CERTAIN TM-147
THE CRYPTOGRAPHIC SECURITY OF COMPACT TM-164
EMBEDDING CRYPTOGRAPHIC TRAPDOORS IN ARBITRARY KNAPSACK TM-230

knowledge based systems
ARTIFICIAL INTELLIGENCE AND CLINICAL PROBLEM SOLVING TM-140
DESIGN CRITERIA FOR A KNOWLEDGE-BASED ENGLISH LANGUAGE TR-146

knowledge representation
AN OVERVIEW OF OWL, A LANGUAGE FOR KNOWLEDGE TM-086
DESCRIPTIONS AND THE SPECIALIZATION OF CONCEPTS TM-101
ROLES, CO-DESCRIPTORS AND THE FORMAL REPRESENTATION OF TM-139
XLMS A LINGUISTIC MEMORY SYSTEM TM-173
BRAND X MANUAL TM-186
AN ABSTRACT MODEL OF A RESEARCH INSTITUTE SIMPLE TR-118
SOME DATA BASE APPLICATIONS OF CONSTRAINT EXPRESSIONS TR-158
REFLECTION AND SEMANTICS IN A PROCEDURAL LANGUAGE TR-272

knowledge-based systems
RECORD OF THE WORKSHOP ON RESEARCH IN OFFICE SEMANTICS TM-195

Knuth-Bendix
DERIVED PAIRS, OVERLAP CLOSURES, AND REWRITE DOMINOES· TR-268

lambda calculus
WHAT IS A MODEL OF THE LAMBDA CALCULUS? TM-171
WHAT IS A MODEL OF THE LAMBDA CALCULUS? EXPANDED TM-201
POWER SET MODELS OF LAMBDA-CALCULUS THEORIES, . TM-207
LAMBDA CALCULUS MODELS OF PROGRAMMING LANGUAGES TR-057
A FORMAL SYSTEM FOR DEFINING THE SYNTAX AND SEMANTICS TR-060
FUNCTIONAL DOMAINS OF APPLICATIVE LANGUAGES TR-136
REFLECTION AND SEMANTICS IN A PROCEDURAL LANGUAGE TR-272

language definition
A MODEL FOR PROCESS REPRESENTATION AND SYNTHESIS TR-087

language extension
A MODEL FOR PROCESS REPRESENTATION AND SYNTHESIS TR-087

language implementation
A LANGUAGE IMPLEMENTATION SYSTEM TR-126

language recognition
TOWARD A COMPUTATIONAL THEORY OF INDIRECT SPEECH ACTS TR-223

language theory
MATHEMATICAL FOUNDATIONS OF FORMAL LANGUAGE THEORY TR-108

language translation
SOME PROBLEMS IN GERMAN TO ENGLISH MACHINE TRANSLATION TR-142

language-oriented machine architecture
A MACHINE ARCHITECTURE TO SUPPORT AN OBJECT-ORIENTED TR-209

largest processing time
ALGORITHMS FOR SCHEDULING TASKS ON UNRELATED PROCESSORS TM-137

latency
REPRESENTATION AND ANALYSIS OF REAL-TIME CONTROL TR-241

layered system building
LSB MANUAL TM-200

layered systems
A SIMPLE AND FLEXIBLE SYSTEM INITIALIZATION MECHANISM TR-180

layered virtual memory
A LAYERED VIRTUAL MEMORY MANAGER TR-177

layout area
LAYOUTS FOR THE SHUFFLE-EXCHANGE GRAPH AND LOWER BOUND TR-274

layout of integrated circuits
THE IMPACT OF LAYER ASSIGNMENT METHODS ON LAYOUT TR-291

learning models
A COMPUTER MODEL OF SIMPLE FORMS OF LEARNING TM-020

learning
ON BATESON'S LOGICAL LEVELS OF LEARNING TM-057
LEARNING STRUCTURAL DESCRIPTION FROM EXAMPLES TR-076

level of multiprogramming
HARDWARE ESTIMATION OF A PROCESS' PRIMARY MEMORY TM-081

lexical retrieval
PHRASE-STRUCTURE PARSING A METHOD FOR TAKING ADVANTAGE TR-296

libraries
BOUNDS ON INFORMATION RETRIEVAL EFFICIENCY IN STATIC TR-088

life of longest dead path	
AN EFFICIENT ALGORITHM FOR DETERMINING THE LENGTH OF	TM-149
LIFO tree search	
AN EFFICIENT ALGORITHM FOR DETERMINING THE LENGTH OF	TM-149
light pens	
INVESTIGATION OF AN ANALOG TECHNIQUE TO DECREASE	TR-025
SYMBOLIC MATHEMATICAL LABORATORY	TR-036
linear dynamic systems	
ON THE SIMULATION OF DYNAMIC SYSTEMS WITH LUD	TR-045
linear programming	
ON LINEAR CHARACTERIZATIONS OF COMBINATORIAL	TM-154
linguistic memory	
XLMS. A LINGUISTIC MEMORY SYSTEM	TM-173
linguistics	
PROCEDURES AS A REPRESENTATION FOR DATA IN A COMPUTER .	TR-084
A FRAMEWORK FOR PROCESSING DIALOGUE	TR-182
LISP	
LOOP ITERATION MACRO	TM-169
BRAND X MANUAL	TM-186
LSB MANUAL	TM-200
MACLISP EXTENSIONS	TM-203
REFLECTION AND SEMANTICS IN A PROCEDURAL LANGUAGE	TR-272
THE REVISED MACLISP MANUAL	TR-295
list memory	
ACTOR SYSTEMS FOR REAL-TIME COMPUTATION	TR-197
list schedules	
EFFICIENT SCHEDULING OF TASKS WITHOUT FULL USE OF	TM-122
list scheduling	
BOUNDS ON THE SCHEDULING OF TYPED TASK SYSTEMS	TM-111
list structure	
THE REVISED MACLISP MANUAL	TR-295
list-processing	
A NEW LIST-TRACING ALGORITHM	TM-019
LMS	
AN OVERVIEW OF OWL, A LANGUAGE FOR KNOWLEDGE	TM-086
local computer networks	
A TELEX GATEWAY FOR THE INTERNET	TM-222
location problems	
WORST-CASE AND PROBABILISTIC ANALYSIS OF A GEOMETRIC	TM-153
locking	
A THEOREM IN DATA BASE CONCURRENCY CONTROL	TM-180
THE COMPLEXITY OF CONCURRENCY CONTROL FOR DISTRIBUTED .	TR-269
log-space reducibility	
ON TIME-SPACE CLASSES AND THEIR RELATION TO THE THEORY	TR-195
logarithmic space	
COMPUTING IN LOGARITHMIC SPACE	TM-052
logic arrays	
AN ASYNCHRONOUS LOGIC ARRAY	TM-062
logic design	
AN ASYNCHRONOUS LOGIC ARRAY	TM-062
REPORT ON THE WORKSHOP ON SELF-TIMED SYSTEMS	TM-166
logic of effective definitions	
A NOTE ON EQUIVALENCES AMONG LOGICS OF PROGRAMS	TM-211
logic simulation	
A SWITCH-LEVEL SIMULATION MODEL FOR INTEGRATED LOGIC	TR-259
logical systems	
COMPLEXITY MEASURES FOR LANGUAGE RECOGNITION BY CANONIC	TR-077
logics of programs	
WEAK MONADIC SECOND ORDER THEORY OF SUCCESSOR IS NOT	TM-038
APPLICATIONS OF MODAL LOGIC TO PROGRAMMING	TM-116
SIX LECTURES ON DYNAMIC LOGIC	TM-117
TEN THOUSAND AND ONE LOGICS OF PROGRAMMING	TM-150
A NOTE ON EQUIVALENCES AMONG LOGICS OF PROGRAMS	TM-211
SEMANTICAL CONSIDERATIONS ON FLOYD-HOARE LOGIC	TR-168
LOGICS OF PROGRAMS AXIOMATICS AND DESCRIPTIVE POWER	TR-200
A SURVEY OF THE LOGIC OF EFFECTIVE DEFINITIONS	TR-246
PROPOSITIONAL DYNAMIC LOGIC OF LOOPING AND CONVERSE	TR-263
logic	
DYNAMIC ALGEBRAS· EXAMPLES, CONSTRUCTIONS, APPLICATIONS	TM-138
DYNAMIC ALGEBRAS AND THE NATURE OF INDUCTION	TM-159
MATHEMATICAL LOGIC FOR COMPUTER SCIENTISTS	TR-131
SOME UPPER AND LOWER BOUNDS ON DECISION PROCEDURES IN .	TR-139
THE COMPUTATIONAL COMPLEXITY OF SOME LOGICAL THEORIES	TR-144
SEMANTICAL CONSIDERATIONS ON FLOYD-HOARE LOGIC	TR-168
THE LOGIC OF SYSTEMS	TR-170
longest common subsequence	
A FASTER ALGORITHM COMPUTING STRING EDIT DISTANCES	TM-105
looping	
ON THE EXPRESSIVE POWER OF DYNAMIC LOGIC	TM-157
lower bounds	
COMBINING DIMENSIONALITY AND RATE OF GROWTH ARGUMENTS .	TM-046
LR(k) languages	
PRACTICAL TRANSLATORS FOR LR(K) LANGUAGES	TR-065
LRU stack model	
HARDWARE ESTIMATION OF A PROCESS' PRIMARY MEMORY .	TM-081
LSB	
LSB MANUAL	TM-200
lurking bugs	
INTERACTIVE DEBUGGING IN A DISTRIBUTED COMPUTATIONAL	TR-264
machine aided cognition	
COMPUTER RECOGNITION OF PRISMATIC SOLIDS	TR-073
machine code generation	
AUTOMATIC CREATION OF A CODE GENERATOR FROM A MACHINE	TR-085
machine descriptions	
A PORTABLE COMPILER FOR THE LANGUAGE C	TR-149
machine perception	
LEARNING STRUCTURAL DESCRIPTION FROM EXAMPLES	TR-076
machine simulation	
TWO REMARKS ON THE POWER OF COUNTING	TM-228
machine understanding	
PROCEDURES AS A REPRESENTATION FOR DATA IN A COMPUTER .	TR-084
machine-aided cognition	
THE MAC-SYSTEM A PROGRESS REPORT	TR-012
AN ANALYSIS OF TIME-SHARED COMPUTER SYSTEMS	TR-018
A HEURISTIC APPROACH TO ALTERNATE ROUTING IN A JOB SHOP	TR-019
QUEUEING MODELS FOR FILE MEMORY OPERATION	TR-021
MODELS AND DATA STRUCTURES FOR DIGITAL LOGIC SIMULATION	TR-031
AN ON-LINE SYSTEM FOR ALGEBRAIC MANIPULATION	TR-035
SOME ASPECTS OF PATTERN RECOGNITION BY COMPUTER	TR-037
machine-independent code generation	
THE SPECIFICATION OF CODE GENERATION ALGORITHMS	TR-199
MACLISP	
LSB MANUAL	TM-200
MACLISP EXTENSIONS	TM-203
THE REVISED MACLISP MANUAL	TR-295
macro control language	
A USER'S GUIDE TO THE MACRO CONTROL LANGUAGE	TM-036
macroprocessor	
AUTOMATIC CODE-GENERATION FROM AN OBJECT-MACHINE	TM-018
Macro	
LOOP ITERATION MACRO	TM-169
MACSYMA	
COMPUTER PROGRAMS FOR RESEARCH IN GRAVITATION AND	TM-167
EVALUATION OF DEFINITE INTEGRALS BY SYMBOLIC	TR-092
ESSAYS IN ALGEBRAIC SIMPLIFICATION	TR-095
MAC	
THE MAC-SYSTEM. A PROGRESS REPORT	TR-012
main memory	
PROGRAM STRUCTURE IN A MULTI-ACCESS COMPUTER	TR-011
man computer interface	
AN EASE OF USE EVALUATION OF AN INTEGRATED EDITOR AND .	TR-266
DISPLAY MANAGEMENT IN AN INTEGRATED OFFICE	TR-278
man-computer interaction	
THE SUBSTANTIVE USE OF COMPUTERS FOR INTELLECTUAL	TM-021
HELPING PEOPLE THINK	TM-025
man-machine decision system	
TRANSMISSION OF INFORMATION BETWEEN A MAN-MACHINE	TM-022
man-machine interaction	
MAP A SYSTEM FOR ON-LINE MATHEMATICAL ANALYSIS	TR-024
PILOT A STEP TOWARDS MAN-COMPUTER SYMBIOSIS	TR-032
management applications	
THE OPS-1 MANUAL	TR-008
management information systems	
THE SUBSTANTIVE USE OF COMPUTERS FOR INTELLECTUAL	TM-021
THE RELATIONAL APPROACH TO THE MANAGEMENT OF DATA BASES	TM-023
THE MACAIMS DATA MANAGEMENT SYSTEM	TM-024
HELPING PEOPLE THINK	TM-025
management support system	
DESIGN CRITERIA FOR A KNOWLEDGE-BASED ENGLISH LANGUAGE	TR-146
Manhattan routing	
AN APPROXIMATION ALGORITHM FOR MANHATTAN ROUTING	TM-238
marked graphs	
THE LOGIC OF SYSTEMS	TR-170
matching	
LEARNING STRUCTURAL DESCRIPTION FROM EXAMPLES	TR-076
mathematics	
CONSTRUCTION HEURISTICS FOR GEOMETRY AND A VECTOR	TM-028
A CLASS OF FINITE COMPUTATION STRUCTURES SUPPORTING THE	TM-031
A DECISION PROCEDURE FOR THE FIRST ORDER THEORY OF REAL	TM-033
POLYNOMIAL EXPONENTIATION THE FAST FOURIER TRANSFORM .	TM-034
AN IMPROVED OVERLAP ARGUMENT FOR ON-LINE MULTIPLICATION	TM-040
ON THE COMPLEXITY OF THE THEORIES OF WEAK DIRECT	TM-042
SUPER-EXPONENTIAL COMPLEXITY OF PRESBURGER ARITHMETIC	TM-043
FAST ON-LINE INTEGER MULTIPLICATION	TM-045
MATHEMATICAL FOUNDATIONS OF FLIP-FLOPS	TM-047
THE REDUCTION METHOD FOR ESTABLISHING LOWER BOUNDS ON ..	TM-048
THE COMPLEXITY OF NEGATION-LIMITED NETWORKS- A BRIEF	TM-065
ON TRIANGULATIONS OF A SET OF POINTS IN THE PLANE	TM-088
FACTORING NUMBERS IN 0 (LOG n) ARITHMETIC STEPS	TM-091
WITH WHAT FREQUENCY ARE APPARENTLY INTRACTABLE PROBLEMS	TM-126

CONTROLLED INFORMATION SHARING IN A COMPUTER UTILITY TR-067
THE CLASSROOM INFORMATION AND COMPUTING SERVICE TR-080

multiplexed computers
THE CLASSROOM INFORMATION AND COMPUTING SERVICE TR-080

multipoint net
AN APPROXIMATION ALGORITHM FOR MANHATTAN ROUTING TM-238

multiprocess computer systems
RESOURCE ALLOCATION IN MULTIPROCESS COMPUTER SYSTEMS TR-050

multiprocessing
SUSPENSION OF PROCESSES IN A MULTIPROCESSING COMPUTER TM-014
PERFORMANCE EVALUATION OF MULTIPROGRAMMED TIME-SHARED TR-103
ACTORS AND CONTINUOUS FUNCTIONALS TR-194

multiprocessor computer systems
RESEARCH DIRECTIONS IN COMPUTER ARCHITECTURE TM-114

multiprocessor scheduling
TASK SCHEDULING IN THE CONTROL ROBOTICS ENVIRONMENT TM-077

multiprocessor systems
INPUT/OUTPUT IN TIME-SHARED, SEGMENTED, MULTIPROCESSOR TR-028
REFERENCE TREE NETWORKS VIRTUAL MACHINE AND . TR-222
THE DESIGN OF A MULTIPROCESSOR DEVELOPMENT SYSTEM TR-279

multiprogramming
PROGRAMMING SEMANTICS FOR MULTIPROGRAMMED COMPUTATIONS TR-023
PERFORMANCE EVALUATION OF MULTIPROGRAMMED TIME-SHARED TR-103

multiset ordering
RECURSIVE DECOMPOSITION ORDERING AND MULTISET ORDERINGS TM-219

music applications
SYNTAX-BASED ANALYTIC READING OF MUSICAL SCORES TR-039

music synthesis
SOME NEW METHODS OF MUSIC SYNTHESIS TM-172

musical synthesis
A PARALLEL PROCESSING MODEL OF MUSICAL STRUCTURES TR-091

mutual exclusion problem
THE MUTUAL EXCLUSION PROBLEM FOR UNRELIABLE PROCESSES TM-084
COORDINATION OF PARALLEL PROCESSES IN THE ACTOR MODEL TR-173

mutual exclusion
SEMAPHORE PRIMITIVES AND STARVATION-FREE MUTUAL . TM-158

mutually suspicious subsystems
COOPERATION OF MUTUALLY SUSPICIOUS SUBSYSTEMS IN A TR-104
NAMING AND PROTECTION IN EXTENDABLE OPERATING SYSTEMS TR-140

natural language
A SYSTEM TO PROCESS DIALOGUE. A PROGRESS REPORT TM-079
AN OVERVIEW OF OWL, A LANGUAGE FOR KNOWLEDGE TM-086
COPING WITH SYNTACTIC AMBIGUITY OR HOW TO PUT THE BLOCK TM-216
NATURAL LANGUAGE INPUT FOR A COMPUTER PROBLEM SOLVING . TR-001
SIR A COMPUTER PROGRAM FOR SEMANTIC INFORMATION TR-002
PROCEDURES AS A REPRESENTATION FOR DATA IN A COMPUTER TR-084
SOME PROBLEMS IN GERMAN TO ENGLISH MACHINE TRANSLATION TR-142
DESIGN CRITERIA FOR A KNOWLEDGE-BASED ENGLISH LANGUAGE TR-146
TOWARD A COMPUTATIONAL THEORY OF INDIRECT SPEECH ACTS TR-223
AUTOMATIC EXTENSION OF AN AUGMENTED TRANSITION NETWORK TR-233
ON MEMORY LIMITATIONS IN NATURAL LANGUAGE PROCESSING TR-245
PRELIMINARY ANALYSIS OF A BREADTH-FIRST PARSING TR-261

negation-limited networks
THE COMPLEXITY OF NEGATION-LIMITED NETWORKS- A BRIEF TM-065

network flow
THE MAX FLOW ALGORITHM OF DINIC AND KARZANOV AN TM-080
CONSTRUCTION AND ANALYSIS OF NETWORK FLOW PROBLEM WHICH TM-083
THE COMPLEXITY OF THE MAXIMUM NETWORK FLOW PROBLEM TR-230

network memory
LEARNING STRUCTURAL DESCRIPTION FROM EXAMPLES TR-076

networks
A NETWORK TRAFFIC GENERATOR FOR DECNET TM-127
REVERSIBLE COMPUTING TM-151
CONSERVATIVE LOGIC TM-197
COMMUNICATION RING INITIALIZATION WITHOUT CENTRAL . TM-202
A TELEX GATEWAY FOR THE INTERNET TM-222
PACKET COMMUNICATION TR-114
FACILITATING INTERPROCESS COMMUNICATION IN A TR-184
ACTORS AND CONTINUOUS FUNCTIONALS TR-194
MULTIPLE-PROCESSOR IMPLEMENTATIONS OF MESSAGE PASSING TR-198
A MINICOMPUTER NETWORK SIMULATION SYSTEM TR-208
REFERENCE TREE NETWORKS VIRTUAL MACHINE AND TR-222
THE EVENT BASED LANGUAGE AND ITS MULTIPLE PROCESSOR TR-226

NIL
LSB MANUAL TM-200
MACLISP EXTENSIONS TM-203

nodes
A GRAPH MODEL FOR PARALLEL COMPUTATIONS TR-064

non-crossreferencing
FURTHER RESULTS ON HIERARCHIES OF CANONIC SYSTEMS TR-100

non-determinacy
FAULT TOLERANCE IN PACKET COMMUNICATION COMPUTER TR-250

non-determinate programs
DENOTATIONAL SEMANTICS OF DETERMINATE AND TR-220

non-inserting
FURTHER RESULTS ON HIERARCHIES OF CANONIC SYSTEMS TR-100

non-numerical analysis
SYMBOLIC MATHEMATICAL LABORATORY TR-036

nondeterminism
NONDETERMINISM IN LOGICS OF PROGRAMS TM-098
CONCURRENT PROGRAMMING TM-115
ON THE EXPRESSIVE POWER OF DYNAMIC LOGIC, II TM-204
SIMULATIONS AMONG MULTIDIMENSIONAL TURING MACHINES TR-242

nonlinear parameter estimation
EFFICIENT MODELING FOR SHORT CHANNEL MOS CIRCUIT TR-277

nonpreemptive schedules
EFFICIENT SCHEDULING OF TASKS WITHOUT FULL USE OF . TM-122
ALGORITHMS FOR SCHEDULING TASKS ON UNRELATED PROCESSORS TM-137

normal-form
REFLECTION AND SEMANTICS IN A PROCEDURAL LANGUAGE TR-272

novice programming
A PRINCIPLED DESIGN FOR AN INTEGRATED COMPUTATIONAL TM-223

NP = P
TIME, SPACE AND RANDOMNESS TM-131

NP and co-NP
THE COMPLEXITY OF EVALUATION RELATIONAL QUERIES TM-229

NP-complete problems
ON THE CRYPTOCOMPLEXITY OF KNAPSACK SYSTEMS TM-129

NP-complete
ON TRIANGULATIONS OF A SET OF POINTS IN THE PLANE TM-088
WITH WHAT FREQUENCY ARE APPARENTLY INTRACTABLE PROBLEMS TM-126
A T = $0(2^{n/2})$, S = $0(2^{n/4})$ ALGORITHM FOR CERTAIN .. TM-147
WORST-CASE AND PROBABILISTIC ANALYSIS OF A GEOMETRIC . TM-153
ON THE COMPUTATIONAL COMPLEXITY OF CARDINALITY TM-160
HAMILTON PATHS IN GRID GRAPHS TM-182
PLANAR EMBEDDING OF PLANAR GRAPHS TM-237
SERIALIZABILITY OF CONCURRENT DATA BASE UPDATES TR-210
THE EVENT BASED LANGUAGE AND ITS MULTIPLE PROCESSOR . TR-226
ALGORITHMS FOR INTEGRATED CIRCUIT LAYOUT AN ANALYTIC . TR-248

null values
ON DATA BASES WITH INCOMPLETE INFORMATION TM-142

numbered sets
RECURSION THEORETIC OPERATORS AND MORPHISMS ON NUMBERED TM-194

numerical analysis
MAP· A SYSTEM FOR ON-LINE MATHEMATICAL ANALYSIS TR-024
EPS AN INTERACTIVE SYSTEM FOR SOLVING ELLIPTIC TR-062

numerical calculations
CALCULAID: AN ON-LINE SYSTEM FOR ALGEBRAIC COMPUTATION TR-020

numerical manipulations
CALCULAID. AN ON-LINE SYSTEM FOR ALGEBRAIC COMPUTATION TR-020

numerical optimization
EFFICIENT MODELING FOR SHORT CHANNEL MOS CIRCUIT TR-277

object identification
SOME ASPECTS OF PATTERN RECOGNITION BY COMPUTER TR-037

object oriented programming
A MACHINE ARCHITECTURE TO SUPPORT AN OBJECT-ORIENTED . TR-209
TRANSMITTING ABSTRACT VALUES IN MESSAGES TR-234

office analysis
EVALUATION OF AN OFFICE ANALYSIS METHODOLOGY TM-239
ANALYSIS AND SPECIFICATION OF OFFICE PROCEDURES TR-275
OFFICE ANALYSIS METHODOLOGY AND CASE STUDIES TR-289
AN OFFICE ANALYSIS AND DIAGNOSIS METHODOLOGY TR-290

office automation
PROGRAMS FOR DISTRIBUTED COMPUTING THE CALENDAR TM-168
RECORD OF THE WORKSHOP ON RESEARCH IN OFFICE SEMANTICS TM-195
SOFTWARE FOR THE 'ROLES' PEOPLE PLAY TM-210
PCAL A PERSONAL CALENDAR TM-213
COOPERATIVE OFFICE WORK, TELECONFERENCING AND CALENDAR TM-218
SUPPORTING ORGANIZATIONAL PROBLEM SOLVING WITH A TM-224
FOUNDATIONS FOR OFFICE SEMANTICS TM-225
EVALUATION OF AN OFFICE ANALYSIS METHODOLOGY TM-239
AN INTEGRATED APPROACH TO FORMATTED DOCUMENT PRODUCTION TR-253
AN EASE OF USE EVALUATION OF AN INTEGRATED EDITOR AND . TR-266
ANALYSIS AND SPECIFICATION OF OFFICE PROCEDURES TR-275
DISPLAY MANAGEMENT IN AN INTEGRATED OFFICE TR-278
OFFICE ANALYSIS METHODOLOGY AND CASE STUDIES TR-289
AN OFFICE ANALYSIS AND DIAGNOSIS METHODOLOGY TR-290

office model
EVALUATION OF AN OFFICE ANALYSIS METHODOLOGY TM-239
OFFICE ANALYSIS METHODOLOGY AND CASE STUDIES TR-289
AN OFFICE ANALYSIS AND DIAGNOSIS METHODOLOGY TR-290

office semantics
SUPPORTING ORGANIZATIONAL PROBLEM SOLVING WITH A . TM-224
FOUNDATIONS FOR OFFICE SEMANTICS TM-225

omega
SUPPORTING ORGANIZATIONAL PROBLEM SOLVING WITH A . TM-224

on-line computer systems
CTSS TECHNICAL NOTES TR-016
AN ANALYSIS OF TIME-SHARED COMPUTER SYSTEMS TR-018

A HEURISTIC APPROACH TO ALTERNATE ROUTING IN A JOB SHOP	TR-019
QUEUEING MODELS FOR FILE MEMORY OPERATION	TR-021
operating systems	
SUSPENSION OF PROCESSES IN A MULTIPROCESSING COMPUTER	TM-014
SIM360 A S/360 SIMULATOR	TM-030
ANCILLARY REPORTS KERNEL DESIGN PROJECT	TM-087
DESIGN STRATEGIES FOR FILE SYSTEMS	TR-078
THE CLASSROOM INFORMATION AND COMPUTING SERVICE	TR-080
DYNAMIC RECONFIGURATION IN A MODULAR COMPUTER SYSTEM	TR-086
PERFORMANCE EVALUATION OF MULTIPROGRAMMED TIME-SHARED	TR-103
NAMING AND PROTECTION IN EXTENDABLE OPERATING SYSTEMS	TR-140
FINAL REPORT OF THE MULTICS KERNEL DESIGN PROJECT	TR-196
DISTRIBUTED COMPUTER SYSTEMS STRUCTURE AND SEMANTICS	TR-215
THE ARCHITECTURE OF AN OBJECT BASED PERSONAL COMPUTER	TR-232
operational semantics	
OPERATIONAL SEMANTICS OF A DATA FLOW LANGUAGE	TM-120
operators	
AN OPERATOR EMBEDDING THEOREM FOR COMPLEXITY CLASSES OF	TM-032
RECURSION THEORETIC OPERATORS AND MORPHISMS ON NUMBERED	TM-194
OPL-I	
OPL-I AN OPEN ENDED PROGRAMMING SYSTEM WITHIN CTSS	TR-007
OPS-1	
THE OPS-1 MANUAL	TR-008
optical disk	
RECOVERY OF THE SWALLOW REPOSITORY	TR-252
optimal algorithms	
OPTIMAL ARRANGEMENT OF KEYS IN A HASH TABLE	TM-073
optimal schedulers	
AN OPTIMALITY THEORY OF CONCURRENCY CONTROL FOR	TM-185
optimization of simulation models	
IN-PROCESS MANUFACTURING QUALITY CONTROL	TR-083
optimization	
SAFETY AND OPTIMIZATION TRANSFORMATIONS FOR DATA FLOW	TR-240
optimizing compilers	
INTERPROCEDURAL DATA FLOW ANALYSIS IN THE PRESENCE OF	TR-247
oracle computation	
TWO REMARKS ON THE POWER OF COUNTING	TM-228
orchestra file	
APPLICATION OF DATA FLOW ARCHITECTURE TO COMPUTER MUSIC	TR-257
orphans	
INTERNAL CONSISTENCY OF A DISTRIBUTED TRANSACTION	TR-286
outerplanar graphs	
PLANAR EMBEDDING OF PLANAR GRAPHS	TM-237
overlap argument	
AN IMPROVED OVERLAP ARGUMENT FOR ON-LINE MULTIPLICATION	TM-040
AN IMPROVED PROOF OF THE RABIN-HARTMANIS-STEARNS	TM-123
overlap closure	
DERIVED PAIRS, OVERLAP CLOSURES, AND REWRITE DOMINOES	TR-268
overlap	
SPACE-BOUNDED SIMULATION OF MULTITAPE TURING MACHINES	TM-148
SIMULATIONS AMONG MULTIDIMENSIONAL TURING MACHINES	TR-242
OWL	
AN OVERVIEW OF OWL, A LANGUAGE FOR KNOWLEDGE	TM-086
DESCRIPTIONS AND THE SPECIALIZATION OF CONCEPTS	TM-101
P and NP	
ON THE COMPLEXITY OF INTEGER PROGRAMMING	TM-152
P, NP, and co-NP	
ON LINEAR CHARACTERIZATIONS OF COMBINATORIAL	TM-154
P, NP, and NP-complete problems	
COMPUTATIONAL COMPLEXITY AND THE TRAVELING SALESMAN	TM-209
P-space hard	
INTERPROCEDURAL DATA FLOW ANALYSIS IN THE PRESENCE OF	TR-247
packet communication	
PACKET COMMUNICATION	TR-114
A STRUCTURE MEMORY FOR DATA FLOW COMPUTERS	TR-186
SIMULATION OF PACKET COMMUNICATION ARCHITECTURE	TR-188
FORMAL SPECIFICATIONS FOR PACKET COMMUNICATION SYSTEMS	TR-189
packing rule	
NEAR-OPTIMAL BIN PACKING ALGORITHMS	TR-109
paging	
FILE MANAGEMENT AND RELATED TOPICS	TM-012
AN EXPERIMENTAL ANALYSIS OF PROGRAM REFERENCE PATTERNS	TR-127
PROGRAM RESTRUCTURING FOR VIRTUAL MEMORY SYSTEMS	TR-148
A MULTI-PROCESS DESIGN OF PAGING SYSTEM	TR-171
PAL	
AN EXPANSION OF THE DATA STRUCTURING CAPABILITIES OF PAL	TM-015
parallel algorithms	
THE FLOW GRAPH SCHEMATA MODEL OF PARALLEL COMPUTATION	TR-053
parallel computation	
THE USE OF QUEUES IN THE PARALLEL DATA FLOW EVALUATION	TM-104
OPTIMIZING SYNCHRONOUS SYSTEMS	TM-215
NEW LOWER BOUND TECHNIQUES FOR VLSI	TM-227
THE FLOW GRAPH SCHEMATA MODEL OF PARALLEL COMPUTATION	TR-053
A GRAPH MODEL FOR PARALLEL COMPUTATIONS	TR-064
HIERARCHICAL ASSOCIATIVE MEMORIES FOR PARALLEL	TR-069
PRODUCTIVITY IN PARALLEL COMPUTATION SCHEMATA	TR-111
A PARALLEL ASYNCHRONOUS COMPUTER ARCHITECTURE FOR DATA	TR-150
ANALYSIS OF THE SIMPLE CODE FOR DATA FLOW COMPUTATION	TR-216
DENOTATIONAL SEMANTICS OF DETERMINATE AND	TR-220
PARALLEL COMPUTATION SYNCHRONIZATION, SCHEDULING, AND	TR-231
LAYOUTS FOR THE SHUFFLE-EXCHANGE GRAPH AND LOWER BOUND	TR-274
parallel processing	
SEMAPHORE PRIMITIVES AND STARVATION-FREE MUTUAL	TM-158
A DATA FLOW ARCHITECTURE WITH TAGGED TOKENS	TM-174
SEMANTICS OF COMMUNICATING PARALLEL PROCESSES	TR-154
COORDINATION OF PARALLEL PROCESSES IN THE ACTOR MODEL	TR-173
THE DESIGN OF A MULTIPROCESSOR DEVELOPMENT SYSTEM	TR-279
AN IMPLEMENTATION SCHEME FOR ARRAY OPERATIONS IN STATIC	TR-280
parallel programming	
STREAM-ORIENTED COMPUTATION IN RECURSIVE DATA FLOW	TM-068
AN INTERMEDIATE FORM FOR DATA FLOW PROGRAMS	TM-143
PROGRAMMING SEMANTICS FOR MULTIPROGRAMMED COMPUTATIONS	TR-023
THE EVENT BASED LANGUAGE AND ITS MLTIPLE PROCESSOR	TR-226
parallel schemata	
THE USE OF QUEUES IN THE PARALLEL DATA FLOW EVALUATION	TM-104
parallel-processing	
COORDINATION OF ASYNCHRONOUS EVENTS	TR-072
parallelism	
A COMPUTER ARCHITECTURE FOR DATA FLOW COMPUTATION	TM-100
I-STRUCTURES AN EFFICIENT DATA TYPE FOR FUNCTIONAL	TM-178
ACTORS AND CONTINUOUS FUNCTIONALS	TR-194
ACTOR SYSTEMS FOR REAL-TIME COMPUTATION	TR-197
parity problems	
TWO REMARKS ON THE POWER OF COUNTING	TM-228
parrallelism	
SPECIFICATION AND VERIFICATION TECHNIQUES FOR PARALLEL	TR-191
parsing	
COPING WITH SYNTACTIC AMBIGUITY OR HOW TO PUT THE BLOCK	TM-216
SYNTAX-BASED ANALYTIC READING OF MUSICAL SCORES	TR-039
PROCEDURES AS A REPRESENTATION FOR DATA IN A COMPUTER	TR-084
A NEW GRAMMATICAL TRANSFORMATION INTO DETERMINISTIC	TR-119
DESIGN CRITERIA FOR A KNOWLEDGE-BASED ENGLISH LANGUAGE	TR-146
ON MEMORY LIMITATIONS IN NATURAL LANGUAGE PROCESSING	TR-245
PRELIMINARY ANALYSIS OF A BREADTH-FIRST PARSING	TR-261
PHRASE-STRUCTURE PARSING A METHOD FOR TAKING ADVANTAGE	TR-296
partial correctness	
A COMPLETE AXIOMATIC SYSTEM FOR PROVING DEDUCTIONS	TM-096
AXIOMATIC DEFINITIONS OF PROGRAMMING LANGUAGES A	TM-163
AXIOMATIC DEFINITIONS OF PROGRAMMING LANGUAGES, II	TM-179
HOARES'S LOGIC IS NOT COMPLETE WHEN IT COULD BE	TM-226
A SURVEY OF THE LOGIC OF EFFECTIVE DEFINITIONS	TR-246
partial orders of events	
ACTOR SYSTEMS FOR REAL-TIME COMPUTATION	TR-197
partially ordered quantifiers	
CHARACTERIZING SECOND ORDER LOGIC WITH FIRST ORDER	TM-095
path algorithms	
OPTIMIZING SYNCHRONOUS SYSTEMS	TM-215
pathfinding algorithms	
THE SUBGRAPH HOMEOMORPHISM PROBLEM	TM-099
pattern matching	
ON THE WORST-CASE OF BEHAVIOR OF STRING-SEARCHING	TM-071
pattern recognition	
SOME ASPECTS OF PATTERN RECOGNITION BY COMPUTER	TR-037
COMPUTER RECOGNITION OF THREE-DIMENSIONAL OBJECTS IN A	TR-059
COMPUTER RECOGNITION OF PRISMATIC SOLIDS	TR-073
COMPUTER ANALYSIS OF VISUAL PROPERTIES OF CURVED OBJECTS	TR-082
pebble game	
MINIMUM REGISTER ALLOCATION IS COMPLETE IN POLYNOMIAL	TM-128
THE SPACE COMPLEXITY OF TWO PEBBLE GAMES ON TREES	TM-133
ON TIME VERSUS SPACE III	TM-175
SIMULATIONS AMONG MULTIDIMENSIONAL TURING MACHINES	TR-242
pen tracking	
INVESTIGATION OF AN ANALOG TECHNIQUE TO DECREASE	TR-025
perception	
COMPUTER RECOGNITION OF PRISMATIC SOLIDS	TR-073
performance analysis	
ANALYSIS OF ASYNCHRONOUS CONCURRENT SYSTEMS BY TIMED	TR-120
performance	
MODELING AND DECOMPOSITION OF INFORMATION SYSTEMS FOR .	TM-026
MEASURING USER CHARACTERISTICS ON THE MULTICS SYSTEM	TM-089
REPORT ON THE WORKSHOP ON DATA FLOW COMPUTER AND	TM-092
ALGORITHMS FOR SCHEDULING TASKS ON UNRELATED PROCESSORS	TM-137
TIMEPAD - A PERFORMANCE IMPROVING SYNCHRONIZATION	TM-177
A SIMULATOR OF MULTIPLE INTERACTIVE USERS TO DRIVE A	TR-058
PERFORMANCE EVALUATION OF MULTIPROGRAMMED TIME-SHARED	TR-103
ON MEMORY LIMITATIONS IN NATURAL LANGUAGE PROCESSING	TR-245
permanent storage	
A MANAGER FOR NAMED, PERMANENT OBJECTS	TM-162

persistence	
AN EFFECTIVE REPRESENTATION OF THE REACHABILITY SET OF .	TM-188
PERSISTENCE OF VECTOR REPLACEMENT SYSTEMS IS DECIDABLE	TM-189
personal computers	
RESEARCH DIRECTIONS IN COMPUTER ARCHITECTURE	TM-114
A FILE TRANSFER PROGRAM FOR A PERSONAL COMPUTER	TM-217
IMPLEMENTING INTERNET REMOTE LOGIN ON A PERSONAL	TM-233
THE ARCHITECTURE OF AN OBJECT BASED PERSONAL COMPUTER	TR-232
COMPUTERS AND PEOPLE PERSONAL COMPUTATION	TR-249
PROTECTING EXTERNALLY SUPPLIED SOFTWARE IN SMALL .	TR-255
personal data base	
PCAL A PERSONAL CALENDAR	TM-213
Petri nets	
DECISION PROBLEMS FOR PETRI NETS AND VECTOR ADDITION	TM-059
AN ASYNCHRONOUS LOGIC ARRAY	TM-062
A LOGIC DESIGN FOR THE CELL BLOCK OF A DATA FLOW	TM-093
AN EFFECTIVE REPRESENTATION OF THE REACHABILITY SET OF	TM-188
PERSISTENCE OF VECTOR REPLACEMENT SYSTEMS IS DECIDABLE	TM-189
THE COMPLEXITY OF THE WORD PROBLEMS FOR COMMUTATIVE .	TM-199
ANALYSIS OF PRODUCTION SCHEMATA BY PETRI NETS	TR-094
ANALYSIS OF ASYNCHRONOUS CONCURRENT SYSTEMS BY TIMED	TR-120
PETRI NET LANGUAGES	TR-159
DECIDABILITY QUESTIONS FOR PETRI NETS	TR-161
THE LOGIC OF SYSTEMS	TR-170
THE COMPLEXITY OF THE FINITE CONTAINMENT PROBLEM FOR	TR-181
SAFETY AND OPTIMIZATION TRANSFORMATIONS FOR DATA FLOW	TR-240
phonotactics	
PHRASE-STRUCTURE PARSING A METHOD FOR TAKING ADVANTAGE	TR-296
physical computing	
REVERSIBLE COMPUTING	TM-151
CONSERVATIVE LOGIC	TM-197
picture abstraction	
COMPUTER RECOGNITION OF PRISMATIC SOLIDS	TR-073
picture modules	
THE COMPUTER CONTROL OF CHANGING PICTURES	TR-135
picture processing	
COMPUTER RECOGNITION OF PRISMATIC SOLIDS	TR-073
pipelining	
OPTIMIZING SYNCHRONOUS SYSTEMS	TM-215
ANALYSIS OF ASYNCHRONOUS CONCURRENT SYSTEMS BY TIMED	TR-120
APPLICATION OF DATA FLOW ARCHITECTURE TO COMPUTER MUSIC .	TR-257
HIGH LEVEL VAL CONSTRUCTS IN A STATIC DATA FLOW MACHINE	TR-262
AN IMPLEMENTATION SCHEME FOR ARRAY OPERATIONS IN STATIC	TR-280
placement and routing	
OPTIMAL PLACEMENT FOR RIVER ROUTING	TM-206
THE IMPACT OF LAYER ASSIGNMENT METHODS ON LAYOUT	TR-291
planar graphs	
PLANAR EMBEDDING OF PLANAR GRAPHS	TM-237
PLANNER	
PROCEDURES AS A REPRESENTATION FOR DATA IN A COMPUTER	TR-084
planning aid	
PLY A SYSTEM OF PLAUSIBILITY INFERENCE WITH A	TM-232
plausibility inference	
PLY A SYSTEM OF PLAUSIBILITY INFERENCE WITH A .	TM-232
poker	
MENTAL POKER	TM-125
polyhedral decision problem	
THE COMPLEXITY OF THE MAXIMUM NETWORK FLOW PROBLEM	TR-230
polynomial evaluation algorithm	
BOUNDS ON POLYNOMIAL EVALUATION ALGORITHMS	TR-098
polynomial exponentiation	
POLYNOMIAL EXPONENTIATION THE FAST FOURIER TRANSFORM .	TM-034
polynomial hierarchy	
TWO REMARKS ON THE POWER OF COUNTING	TM-228
polynomial ideal	
THE COMPLEXITY OF THE WORD PROBLEMS FOR COMMUTATIVE	TM-199
polynomial operation	
THE HENSEL LEMMA IN ALGEBRAIC MANIPULATION	TR-138
polynomial space complete	
MINIMUM REGISTER ALLOCATION IS COMPLETE IN POLYNOMIAL .	TM-128
THE PROPOSITIONAL DYNAMIC LOGIC OF DETERMINISTIC,	TM-198
polynomial-time	
WITH WHAT FREQUENCY ARE APPARENTLY INTRACTABLE PROBLEMS	TM-126
COMPUTATIONAL COMPLEXITY AND THE TRAVELING SALESMAN .	TM-209
THE COMPLEXITY OF EVALUATION RELATIONAL QUERIES	TM-229
TRANSLATING UPDATES OF RELATIONAL DATA BASE VIEWS	TR-284
ON COMPUTING GALOIS GROUPS AND ITS APPLICATION TO	TR-288
polynomials	
COMBINING DIMENSIONALITY AND RATE OF GROWTH ARGUMENTS	TM-046
portable compilers	
A PORTABLE COMPILER FOR THE LANGUAGE C	TR-149
power set models	
POWER SET MODELS OF LAMBDA-CALCULUS THEORIES,	TM-207
pragmatics	
TOWARD A COMPUTATIONAL THEORY OF INDIRECT SPEECH ACTS	TR-223
predicate calculus	
APPLICATIONS OF MODAL LOGIC TO PROGRAMMING	TM-116
SIX LECTURES ON DYNAMIC LOGIC	TM-117
preemption	
AN ANALYSIS OF PREEMPTIVE MULTIPROCESSOR JOB SCHEDULING	TM-110
preemptive schedules	
EFFICIENT SCHEDULING OF TASKS WITHOUT FULL USE OF	TM-122
preliminary implementation	
AUTOMATIC SYNTHESIS OF IMPLEMENTATIONS FOR ABSTRACT	TR-276
Presburger arithmetic	
SUPER-EXPONENTIAL COMPLEXITY OF PRESBURGER ARITHMETIC	TM-043
primality	
AN ANALYSIS OF THE SOLOVAY AND STRASSEN TEST FOR	TM-108
primary memory requirement	
HARDWARE ESTIMATION OF A PROCESS' PRIMARY MEMORY .	TM-081
prime numbers	
A METHOD FOR OBTAINING SIGNATURES AND PUBLIC-KEY	TM-082
FACTORING NUMBERS IN 0 (LOG n) ARITHMETIC STEPS	TM-091
priority problem	
THE PRIORITY PROBLEM	TR-022
priority scheduling	
REAL-TIME CONTROL STRUCTURES FOR BLOCK DIAGRAM SCHEMATA	TR-204
privacy	
A METHOD FOR OBTAINING SIGNATURES AND PUBLIC-KEY	TM-082
PRIMARY ACCESS CONTROL IN LARGE-SCALE TIME-SHARED	TR-089
probabilistic algorithms	
AN ANALYSIS OF THE SOLOVAY AND STRASSEN TEST FOR	TM-108
WORST-CASE AND PROBABILISTIC ANALYSIS OF A GEOMETRIC	TM-153
PROBABILISTIC ALGORITHMS IN FINITE FIELDS	TR-213
probabilistic analysis	
WAFER-SCALE INTEGRATION OF SYSTOLIC ARRAYS	TM-236
probabilistic bounds	
ON BISECTING RANDOM GRAPHS	TR-287
probabilistic constructions	
ON CONCENTRATION AND CONNECTION NETWORKS	TM-196
probabilistic inference	
PLY A SYSTEM OF PLAUSIBILITY INFERENCE WITH A	TM-232
probability laws	
PSEUDO-RANDOM SEQUENCES	TM-016
problem solving	
ARTIFICIAL INTELLIGENCE AND CLINICAL PROBLEM SOLVING	TM-140
SUPPORTING ORGANIZATIONAL PROBLEM SOLVING WITH A .	TM-224
FOUNDATIONS FOR OFFICE SEMANTICS	TM-225
procedural reflection	
REFLECTION AND SEMANTICS IN A PROCEDURAL LANGUAGE	TR-272
procedure variables	
INTERPROCEDURAL DATA FLOW ANALYSIS IN THE PRESENCE OF .	TR-247
procedures	
FIRST VERSION OF A DATA FLOW PROCEDURE LANGUAGE	TM-061
process control	
THE DESIGN OF A MODULAR LABORATORY FOR CONTROL ROBOTICS	TM-074
TASK SCHEDULING IN THE CONTROL ROBOTICS ENVIRONMENT	TM-077
process initiation	
A SECURE AND FLEXIBLE MODEL OF PROCESS INITIATION FOR A	TR-163
process notion	
A MODEL FOR PROCESS REPRESENTATION AND SYNTHESIS	TR-087
process-control primitives	
IMPLEMENTING MULTI-PROCESS PRIMITIVES IN A MULTIPLEXED	TR-055
processing dialogue	
A FRAMEWORK FOR PROCESSING DIALOGUE	TR-182
processing units	
PROGRAM STRUCTURE IN A MULTI-ACCESS COMPUTER	TR-011
processor design	
A LOGIC DESIGN FOR THE CELL BLOCK OF A DATA FLOW	TM-093
processor multiplexing	
PROCESSOR MULTIPLEXING IN A LAYERED OPERATING SYSTEM	TR-164
procurement systems	
A PROGRAM FOR THE DESIGN OF PROCUREMENT SYSTEMS	TR-160
production schemata	
ANALYSIS OF PRODUCTION SCHEMATA BY PETRI NETS	TR-094
program analysis	
PROGRAM ANALYSIS BY DIGITAL COMPUTER	TR-043
program correctness	
EFFECTIVENESS	TM-109
program decomposition	
A DATA FLOW ARCHITECTURE WITH TAGGED TOKENS	TM-174
program equations	
COMPLEXITY MEASURES FOR PROGRAMMING LANGUAGES	TM-017
program equivalence relations	
COMPLEXITY MEASURES FOR PROGRAMMING LANGUAGES	TM-017
program graphs	
FIRST VERSION OF A DATA FLOW PROCEDURE LANGUAGE	TM-061
A MACHINE LANGUAGE INSTRUCTION SET FOR A DATA FLOW	TM-146
A GRAPH MODEL FOR PARALLEL COMPUTATIONS	TR-064

program resource usage	
COMPLEXITY MEASURES FOR PROGRAMMING LANGUAGES	TM-017
program restructuring	
PROGRAM RESTRUCTURING FOR VIRTUAL MEMORY SYSTEMS	TR-148
program schemata	
PRODUCTIVITY IN PARALLEL COMPUTATION SCHEMATA	TR-111
program semantics	
A COMPLETENESS RESULT FOR A PROPOSITIONAL DYNAMIC LOGIC	TM-106
program translation	
SEMIAUTOMATIC TRANSLATION OF COBOL INTO HIBOL	TR-256
program understanding	
SEMIAUTOMATIC TRANSLATION OF COBOL INTO HIBOL	TR-256
program verification	
FINDING MINIMUM CUTSETS IN REDUCIBLE GRAPHS	TM-085
A NEAR-OPTIMAL METHOD FOR REASONING ABOUT ACTION	TM-113
DYNAMIC ALGEBRAS EXAMPLES, CONSTRUCTIONS, APPLICATIONS	TM-138
DYNAMIC ALGEBRAS AND THE NATURE OF INDUCTION	TM-159
ANALYSIS OF ALGORITHM IMPLEMENTATIONS	TR-130
programmable logic	
AUTOMATIC TEST, CONFIGURATION AND REPAIR OF CELLULAR ..	TR-151
programming environment	
A ROBUST ENVIRONMENT FOR PROGRAM DEVELOPMENT	TR-175
programming language semantics	
SEMANTICS OF DATA STRUCTURES AND REFERENCES	TR-134
A DENOTATIONAL SEMANTICS OF CLU	TR-201
programming languages	
USE OF HIGH LEVEL LANGUAGES FOR SYSTEMS PROGRAMMING	TM-013
AN EXPANSION OF THE DATA STRUCTURING CAPABILITIES OF PAL	TM-015
COMPLEXITY MEASURES FOR PROGRAMMING LANGUAGES	TM-017
AN INVESTIGATION OF CURRENT LANGUAGE SUPPORT FOR THE ..	TM-051
A DYNAMIC DEBUGGING SYSTEM FOR MDL	TM-094
REPORT ON THE SECOND WORKSHOP ON DATA FLOW COMPUTER AND ...	TM-136
AXIOMATIC DEFINITIONS OF PROGRAMMING LANGUAGES, II	TM-179
BRAND X MANUAL	TM-186
A PRINCIPLED DESIGN FOR AN INTEGRATED COMPUTATIONAL ...	TM-223
COMPUTER DESIGN FOR ASYNCHRONOUSLY REPRODUCIBLE .	TR-034
LAMBDA CALCULUS MODELS OF PROGRAMMING LANGUAGES	TR-057
A FORMAL SYSTEM FOR DEFINING THE SYNTAX AND SEMANTICS ...	TR-060
PRACTICAL TRANSLATORS FOR LR(K) LANGUAGES	TR-065
AUTOMATIC CREATION OF A CODE GENERATOR FROM A MACHINE ...	TR-085
A MODEL FOR PROCESS REPRESENTATION AND SYNTHESIS	TR-087
ON THE DESIGN AND SPECIFICATION OF A COMMON BASE ...	TR-101
A LANGUAGE IMPLEMENTATION SYSTEM	TR-126
THE BCPL REFERENCE MANUAL	TR-141
A FORMALIZATION AND CORRECTNESS PROOF OF THE CGOL ..	TR-147
PETRI NET LANGUAGES	TR-159
A FORMAL DEFINITION OF CLU	TR-193
DISTRIBUTED COMPUTER SYSTEMS: STRUCTURE AND SEMANTICS	TR-215
VAL--A VALUE-ORIENTED ALGORITHMIC LANGUAGE, PRELIMINARY ...	TR-218
CLU REFERENCE MANUAL	TR-225
THE EVENT BASED LANGUAGE AND ITS MULTIPLE PROCESSOR ..	TR-226
AN ABSTRACT IMPLEMENTATION FOR A GENERALIZED DATA FLOW ..	TR-228
TRANSMITTING ABSTRACT VALUES IN MESSAGES	TR-234
TOWARDS A THEORY FOR ABSTRACT DATA TYPES	TR-237
A VERY HIGH LEVEL LANGUAGE FOR BUSINESS DATA PROCESSING	TR-254
THE MDL PROGRAMMING LANGUAGE PRIMER	TR-292
THE MDL PROGRAMMING LANGUAGE	TR-293
THEH MDL PROGRAMMING ENVIRONMENT	TR-294
programming linguistics	
COST ANALYSIS OF DEBUGGING SYSTEMS	TR-090
programming methodology	
A FORMAL DEFINITION OF CLU	TR-193
SYNCHRONIZATION MECHANISMS FOR MODULAR PROGRAMMING ..	TR-211
TRANSMITTING ABSTRACT VALUES IN MESSAGES	TR-234
programming semantics	
PROGRAMMING SEMANTICS FOR MULTIPROGRAMMED COMPUTATIONS	TR-023
programming system	
THE BINDING MODEL A SEMANTIC BASE FOR MODULAR .	TR-145
project management	
USE OF HIGH LEVEL LANGUAGES FOR SYSTEMS PROGRAMMING	TM-013
project scheduling	
DEADLOCK-FREE SHARING OF RESOURCES IN ASYNCHRONOUS ...	TR-075
project-join expressions	
ALGEBRAIC DEPENDENCIES	TM-193
projective view	
TRANSLATING UPDATES OF RELATIONAL DATA BASE VIEWS	TR-284
PROLOG	
REFLECTION AND SEMANTICS IN A PROCEDURAL LANGUAGE	TR-272
proofs about programs	
INDUCTION IN PROOFS ABOUT PROGRAMS	TR-093
proofs of correctness	
A FORMALIZATION OF THE STATE MACHINE SPECIFICATION .	TR-202
propositional dynamic logic	
DETERMINISTIC PROPOSITIONAL DYNAMIC LOGIC· FINITE .	TM-190
PROPOSITIONAL DYNAMIC LOGICS OF PROGRAMS: A SURVEY	TM-191
THE DEDUCIBILITY PROBLEM IN PROPOSITIONAL DYNAMIC LOGIC	TM-192
THE PROPOSITIONAL DYNAMIC LOGIC OF DETERMINISTIC, ...	TM-198
LOGICS OF PROGRAMS. AXIOMATICS AND DESCRIPTIVE POWER	TR-200
PROPOSITIONAL DYNAMIC LOGIC OF LOOPING AND CONVERSE	TR-263
proprietary services	
MAKING COMPUTERS KEEP SECRETS	TR-115
proprietary software	
PROTECTING EXTERNALLY SUPPLIED SOFTWARE IN SMALL ...	TR-255
protection	
TIMESTAMPS AND CAPABILITY-BASED PROTECTION IN A ...	TM-135
COOPERATION OF MUTUALLY SUSPICIOUS SUBSYSTEMS IN A ...	TR-104
MAKING COMPUTERS KEEP SECRETS	TR-115
AN INPUT/OUTPUT ARCHITECTURE FOR VIRTUAL MEMORY ...	TR-117
REMOVING THE DYNAMIC LINKER FROM THE SECURITY KERNEL OF ...	TR-132
NAMING AND PROTECTION IN EXTENDABLE OPERATING SYSTEMS	TR-140
ENCRYPTION-BASED PROTECTION PROTOCOLS FOR INTERACTIVE ...	TR-162
A SECURE AND FLEXIBLE MODEL OF PROCESS INITIATION FOR A ...	TR-163
NON-DISCRETIONARY ACCESS CONTROL FOR DECENTRALIZED ...	TR-179
FINAL REPORT OF THE MULTICS KERNEL DESIGN PROJECT	TR-196
PROTECTING EXTERNALLY SUPPLIED SOFTWARE IN SMALL ...	TR-255
Protosystem I	
AUTOMATIC DESIGN OF DATA PROCESSING SYSTEMS	TM-070
PROTOSYSTEM I: AN AUTOMATIC PROGRAMMING SYSTEM PROTOTYPE	TM-072
pseudopolynomial algorithms	
ON THE COMPLEXITY OF INTEGER PROGRAMMING	TM-152
PSPACE-complete	
THE COMPLEXITY OF CONCURRENCY CONTROL FOR DISTRIBUTED ...	TR-269
psychological reality	
ON MEMORY LIMITATIONS IN NATURAL LANGUAGE PROCESSING	TR-245
public-key cryptosystems	
A METHOD FOR OBTAINING SIGNATURES AND PUBLIC-KEY ...	TM-082
A FAST SIGNATURE SCHEME	TM-107
ON THE SECURITY OF THE MERKLE-HELLMAN CRYPTOGRAPHIC ...	TM-119
DIGITALIZED SIGNATURES AND PUBLIC-KEY FUNCTIONS AS ...	TR-212
PROTECTING EXTERNALLY SUPPLIED SOFTWARE IN SMALL ...	TR-255
pumping lemmas	
PUMPING LEMMAS FOR REGULAR SETS	TM-170
quadratic functions	
THE COMPLEXITY OF MONOTONE BOOLEAN FUNCTIONS AND AN ...	TR-238
quadratic residue	
AN ANALYSIS OF THE SOLOVAY AND STRASSEN TEST FOR ...	TM-108
quality control	
IN-PROCESS MANUFACTURING QUALITY CONTROL	TR-083
quantification theory	
MATHEMATICAL LOGIC FOR COMPUTER SCIENTISTS	TR-131
quantifiers	
ROLES, CO-DESCRIPTORS AND THE FORMAL REPRESENTATION OF ...	TM-139
query language semantics	
ON DATA BASES WITH INCOMPLETE INFORMATION	TM-142
queueing models	
QUEUEING MODELS FOR FILE MEMORY OPERATION	TR-021
queues	
THE USE OF QUEUES IN THE PARALLEL DATA FLOW EVALUATION ...	TM-104
r.e. programs	
COMPUTABILITY AND COMPLETENESS IN LOGICS OF PROGRAMS	TM-097
THE EQUIVALENCE OF R. E. PROGRAMS AND DATA FLOW SCHEMES	TM-121
Rabin-Hartmanis-Sterns conjecture	
AN IMPROVED PROOF OF THE RABIN-HARTMANIS-STEARNS ..	TM-123
random graphs	
ONBISECTING RANDOM GRAPHS	TR-287
randomized encryption	
RANDOMIZED ENCRYPTION TECHNIQUES	TM-234
randomness	
A CONCEPT OF INDEPENDENCE WITH APPLICATIONS IN VARIOUS ...	TR-235
rate of growth	
COMBINING DIMENSIONALITY AND RATE OF GROWTH ARGUMENTS ...	TM-046
THE REDUCTION METHOD FOR ESTABLISHING LOWER BOUNDS ON ..	TM-048
rate problems	
CARPS, A PROGRAM WHICH SOLVES CALCULUS WORD PROBLEMS	TR-051
rational functions	
COMBINING DIMENSIONALITY AND RATE OF GROWTH ARGUMENTS ...	TM-046
THE REDUCTION METHOD FOR ESTABLISHING LOWER BOUNDS ON ...	TM-048
real addition	
A DECISION PROCEDURE FOR THE FIRST ORDER THEORY OF REAL ...	TM-033
real time language recognition	
AN IMPROVED PROOF OF THE RABIN-HARTMANIS-STEARNS ...	TM-123
real-time performance	
APPLICATION OF DATA FLOW ARCHITECTURE TO COMPUTER MUSIC ..	TR-257
real-time scheduling	
REAL-TIME CONTROL STRUCTURES FOR BLOCK DIAGRAM SCHEMATA	TR-204
FUNDAMENTAL DESIGN PROBLEMS OF DISTRIBUTED SYSTEMS FOR ...	TR-297
real-time systems	
THE DESIGN OF A MODULAR LABORATORY FOR CONTROL ROBOTICS	TM-074
TASK SCHEDULING IN THE CONTROL ROBOTICS ENVIRONMENT	TM-077
A HEURISTIC APPROACH TO ALTERNATE ROUTING IN A JOB SHOP	TR-019

security kernel	
ANCILLARY REPORTS: KERNEL DESIGN PROJECT	TM-087
MINIMIZING THE NAMING FACILITIES REQUIRING PROTECTION ...	TR-156
PROCESSOR MULTIPLEXING IN A LAYERED OPERATING SYSTEM	TR-164
A ROBUST ENVIRONMENT FOR PROGRAM DEVELOPMENT	TR-175
FINAL REPORT OF THE MULTICS KERNEL DESIGN PROJECT	TR-196
security	
A METHOD FOR OBTAINING SIGNATURES AND PUBLIC-KEY ...	TM-082
OCAS - ON-LINE CRYPTANALYTIC AID SYSTEM	TR-027
MAKING COMPUTERS KEEP SECRETS	TR-115
COMPUTER AND DATA SECURITY· A COMPREHENSIVE ANNOTATED . .	TR-122
REMOVING THE DYNAMIC LINKER FROM THE SECURITY KERNEL OF ...	TR-132
ENCRYPTION-BASED PROTECTION PROTOCOLS FOR INTERACTIVE ...	TR-162
A SECURE AND FLEXIBLE MODEL OF PROCESS INITIATION FOR A ...	TR-163
NON-DISCRETIONARY ACCESS CONTROL FOR DECENTRALIZED .	TR-179
FINAL REPORT OF THE MULTICS KERNEL DESIGN PROJECT	TR-196
DIGITALIZED SIGNATURES AND PUBLIC-KEY FUNCTIONS AS ...	TR-212
Segerberg axioms	
DYNAMIC ALGEBRAS AND THE NATURE OF INDUCTION	TM-159
segmented computer buses	
THE DESIGN OF A MULTIPROCESSOR DEVELOPMENT SYSTEM	TR-279
segmented	8
FILE MANAGEMENT AND RELATED TOPICS	TM-012
INPUT/OUTPUT IN TIME-SHARED, SEGMENTED, MULTIPROCESSOR ...	TR-028
selection problems	
ON LOWER BOUNDS FOR SELECTION PROBLEMS	TR-121
self dual codes	
COMPLETE CLASSIFICATION OF (24,12) AND (22,11) . .	TM-049
self-reference	
REFLECTION AND SEMANTICS IN A PROCEDURAL LANGUAGE	TR-272
self-timed systems	
REPORT ON THE WORKSHOP ON SELF-TIMED SYSTEMS	TM-166
CIRCUIT ANALYSIS OF SELF-TIMED ELEMENTS FOR NMOS VLSI .	TM-220
FAULT TOLERANCE IN PACKET COMMUNICATION COMPUTER .	TR-250
A DESIGN METHODOLOGY FOR SELF-TIMED SYSTEMS	TR-258
semantic definition	
COST ANALYSIS OF DEBUGGING SYSTEMS	TR-090
semantic graph data model	
DATA MODEL EQUIVALENCE	TM-118
DATA MODEL EQUIVALENCE	TM-118
EQUIVALENCE PROPERTIES OF SEMANTIC DATA MODELS FOR DATA ...	TR-206
semantic information retrieval	
SIR: A COMPUTER PROGRAM FOR SEMANTIC INFORMATION ...	TR-002
semantic models for programming languages	
SEMANTICS OF DATA STRUCTURES AND REFERENCES	TR-134
semantic network	
A SYSTEM TO PROCESS DIALOGUE: A PROGRESS REPORT	TM-079
ROLES, CO-DESCRIPTORS AND THE FORMAL REPRESENTATION OF ...	TM-139
XLMS: A LINGUISTIC MEMORY SYSTEM	TM-173
BRAND X MANUAL	TM-186
semantic relation data model	
DATA MODEL EQUIVALENCE	TM-118
EQUIVALENCE PROPERTIES OF SEMANTIC DATA MODELS FOR DATA ...	TR-206
semantics of parallelism	
ACTOR SYSTEMS FOR REAL-TIME COMPUTATION	TR-197
semantics of programming languages	
SPECIFYING THE SEMANTICS OF WHILE-PROGRAMS: A TUTORIAL ...	TM-130
AXIOMATIC DEFINITIONS OF PROGRAMMING LANGUAGES: A ...	TM-163
TERMINATIONS FOR RECURSIVE PROGRAMS: COMPLETENESS AND ...	TM-214
semantics of programs	
SIX LECTURES ON DYNAMIC LOGIC	TM-117
semantics	
DESCRIPTIONS AND THE SPECIALIZATION OF CONCEPTS	TM-101
AXIOMATIC DEFINITIONS OF PROGRAMMING LANGUAGES, II	TM-179
A FORMAL SYSTEM FOR DEFINING THE SYNTAX AND SEMANTICS ...	TR-060
PROCEDURES AS A REPRESENTATION FOR DATA IN A COMPUTER ...	TR-084
SEMANTICS OF COMMUNICATING PARALLEL PROCESSES	TR-154
HIGH LEVEL EXPRESSION OF SEMANTIC INTEGRITY ..	TR-165
DENOTATIONAL SEMANTICS OF DETERMINATE AND ...	TR-220
TOWARD A COMPUTATIONAL THEORY OF INDIRECT SPEECH ACTS	TR-223
REFLECTION AND SEMANTICS IN A PROCEDURAL LANGUAGE	TR-272
semaphores	
CONCURRENT PROGRAMMING	TM-115
SEMAPHORE PRIMITIVES AND STARVATION-FREE MUTUAL ...	TM-158
A THEOREM IN DATA BASE CONCURRENCY CONTROL	TM-180
semilinear sets	
SEMILINEAR SETS AND APPLICATIONS	TR-183
sequential tests	
PSEUDO-RANDOM SEQUENCES	TM-016
serializability	
SERIALIZABILITY OF CONCURRENT DATA BASE UPDATES	TR-210
MULTILEVEL ATOMICITY - A NEW CORRECTNESS CRITERION FOR ...	TR-281
CONCURRENCY CONTROL FOR RESILIENT NESTED TRANSACTIONS	TR-285
INTERNAL CONSISTENCY OF A DISTRIBUTED TRANSACTION ...	TR-286
serializers	
AUTOMATIC VERIFICATION OF SERIALIZES	TR-229
servers	
THE DESIGN OF A ROUTING SERVICE FOR CAMPUS-WIDE ...	TR-270
set-theoretical data structures	
THE RELATIONAL APPROACH TO THE MANAGEMENT OF DATA BASES	TM-023
THE MACAIMS DATA MANAGEMENT SYSTEM	TM-024
sharing	
COPYING COMPLEX STRUCTURES IN A DISTRIBUTED SYSTEM	TR-219
HIGH LEVEL VAL CONSTRUCTS IN A STATIC DATA FLOW MACHINE	TR-262
shortest paths	
THE COMPLEXITY OF MONOTONE BOOLEAN FUNCTIONS AND AN ...	TR-238
shuffle-exchange graph	
LAYOUTS FOR THE SHUFFLE-EXCHANGE GRAPH BASED ON THE ...	TM-221
AN ASYMPTOTICALLY OPTIMAL LAYOUT FOR THE .	TM-231
LAYOUTS FOR THE SHUFFLE-EXCHANGE GRAPH AND LOWER BOUND . .	TR-274
side effects simplification	
REFLECTION AND SEMANTICS IN A PROCEDURAL LANGUAGE	TR-272
side effects	
ABSTRACT MODEL SPECIFICATIONS FOR DATA ABSTRACTIONS	TR-221
simple canonic systems	
FURTHER RESULTS ON HIERARCHIES OF CANONIC SYSTEMS	TR-100
simplification	
RECURSIVE DECOMPOSITION ORDERING AND MULTISET ORDERINGS	TM-219
simulation as an adjunct to analysis	
IN-PROCESS MANUFACTURING QUALITY CONTROL	TR-083
simulation methodology	
THE OPS-1 MANUAL	TR-008
simulation systems	
INCREMENTAL SIMULATION ON A TIME-SHARED COMPUTER	TR-048
simulation	
REPORT ON THE WORKSHOP ON DATA FLOW COMPUTER AND ...	TM-092
A NEW METHODOLOGY FOR COMPUTER SIMULATION	TR-013
ON THE SIMULATION OF DYNAMIC SYSTEMS WITH LUD ...	TR-045
SIMULATIONS AMONG MULTIDIMENSIONAL TURING MACHINES	TR-242
simulators	
A SIMULATOR OF MULTIPLE INTERACTIVE USERS TO DRIVE A ..	TR-058
social applications	
ON BATESON'S LOGICAL LEVELS OF LEARNING	TM-057
social science research	
ON-LINE ANALYSIS FOR SOCIAL SCIENTISTS	TR-040
societal implications of computers	
INCOMPREHENSIBLE COMPUTER SYSTEMS: KNOWLEDGE WITHOUT ...	TR-227
COMPUTERS AND PEOPLE: PERSONAL COMPUTATION	TR-249
software design	
DATA DRIVEN LOOPS	TR-244
software engineering	
FUNDAMENTAL DESIGN PROBLEMS OF DISTRIBUTED SYSTEMS FOR ...	TR-297
solvability by radicals	
ON COMPUTING GALOIS GROUPS AND ITS APPLICATION TO ...	TR-288
sorting networks	
AN ANALYSIS OF SORTING NETWORKS	TR-105
sound synthesis	
APPLICATION OF DATA FLOW ARCHITECTURE TO COMPUTER MUSIC ...	TR-257
sound-synthesis	
A PARALLEL PROCESSING MODEL OF MUSICAL STRUCTURES	TR-091
space complexity	
SPACE-BOUNDED SIMULATION OF MULTITAPE TURING MACHINES	TM-148
ON TIME VERSUS SPACE III	TM-175
spanning tree	
WAFER-SCALE INTEGRATION OF SYSTOLIC ARRAYS	TM-236
spatial metaphor	
A PRINCIPLED DESIGN FOR AN INTEGRATED COMPUTATIONAL ..	TM-223
specification languages	
ANALYSIS AND SPECIFICATION OF OFFICE PROCEDURES	TR-275
specification	
REPORT ON THE WORKSHOP ON DATA FLOW COMPUTER AND ..	TM-092
ABSTRACT MODEL SPECIFICATIONS FOR DATA ABSTRACTIONS	TR-221
AUTOMATIC VERIFICATION OF SERIALIZES	TR-229
speech acts	
TOWARD A COMPUTATIONAL THEORY OF INDIRECT SPEECH ACTS	TR-223
speech recognition	
PHRASE-STRUCTURE PARSING: A METHOD FOR TAKING ADVANTAGE ...	TR-296
speed independent	
REPORT ON THE WORKSHOP ON SELF-TIMED SYSTEMS	TM-166
A DESIGN METHODOLOGY FOR SELF-TIMED SYSTEMS	TR-258
Spiras algorithm	
A NOTE ON THE AVERAGE TIME TO COMPUTE TRANSITIVE ...	TM-076
stack based languages	
ABSTRACT DATA TYPES IN STACK BASED LANGUAGES	TR-190
starvation	
SYNTHESIS OF SYNCHRONIZATION CODE FOR DATA ABSTRACTIONS	TR-203
state graphs	
THE LOGIC OF SYSTEMS	TR-170

state machine specifications	
A FORMALIZATION OF THE STATE MACHINE SPECIFICATION .	TR-202
static data flow machine	
HIGH LEVEL VAL CONSTRUCTS IN A STATIC DATA FLOW MACHINE	TR-262
static scoping	
REFLECTION AND SEMANTICS IN A PROCEDURAL LANGUAGE	TR-272
statistical decision theory	
COMPUTER RECOGNITION OF PRISMATIC SOLIDS	TR-073
statistics	
ON-LINE ANALYSIS FOR SOCIAL SCIENTISTS	TR-040
Stirling's formula	
THE TRAVELING SALESMAN PROBLEM WITH MANY VISITS TO FEW ..	TM-208
storage allocation	
A NEW LIST-TRACING ALGORITHM	TM-019
storage hierarchy	
STORAGE HIERARCHY SYSTEMS	TR-107
storage management	
I-STRUCTURES: AN EFFICIENT DATA TYPE FOR FUNCTIONAL .	TM-178
ACTOR SYSTEMS FOR REAL-TIME COMPUTATION	TR-197
MANAGEMENT OF OBJECT HISTORIES IN THE SWALLOW REPOSITORY	TR-243
storage modification machines	
REAL-TIME SIMULATION OF MULTIDIMENSIONAL TURING .	TM-037
ON TIME VERSUS SPACE III	TM-175
storage	
A REAL TIME GARBAGE COLLECTOR THAT CAN RECOVER .	TM-184
STORAGE HIERARCHY SYSTEMS	TR-107
STORAGE AND ACCESS COSTS FOR IMPLEMENTATIONS OF . .	TR-217
strategic selection	
STRATEGY SELECTION IN MEDICAL DIAGNOSIS	TR-153
streams	
APPLICATION OF DATA FLOW ARCHITECTURE TO COMPUTER MUSIC .	TR-257
HIGH LEVEL VAL CONSTRUCTS IN A STATIC DATA FLOW MACHINE	TR-262
STRESS	
STRESS: A PROBLEM-ORIENTED LANGUAGE FOR STRUCTURAL ..	TR-006
string editing	
A FASTER ALGORITHM COMPUTING STRING EDIT DISTANCES	TM-105
string-matching	
STRING-MATCHING AND OTHER PRODUCTS	TM-041
string-searching	
ON THE WORST-CASE OF BEHAVIOR OF STRING-SEARCHING .	TM-071
strong type checking	
CLU REFERENCE MANUAL	TR-225
strongly typed objects	
COPYING COMPLEX STRUCTURES IN A DISTRIBUTED SYSTEM	TR-219
structure analysis	
INTERACTIVE DESIGN COORDINATION FOR THE BUILDING	TM-010
STRESS: A PROBLEM-ORIENTED LANGUAGE FOR STRUCTURAL .	TR-006
structure memory	
A STRUCTURE MEMORY FOR DATA FLOW COMPUTERS	TR-186
DATA-STRUCTURING OPERATIONS IN CONCURRENT COMPUTATIONS	TR-224
structured programming	
AN INVESTIGATION OF CURRENT LANGUAGE SUPPORT FOR THE	TM-051
THE ARCHITECTURE OF AN OBJECT BASED PERSONAL COMPUTER	TR-232
REFLECTION AND SEMANTICS IN A PROCEDURAL LANGUAGE	TR-272
Student	
NATURAL LANGUAGE INPUT FOR A COMPUTER PROBLEM SOLVING ..	TR-001
super-exponentiatial complexity	
SUPER-EXPONENTIAL COMPLEXITY OF PRESBURGER ARITHMETIC	TM-043
supercomputers	
RESEARCH DIRECTIONS IN COMPUTER ARCHITECTURE	TM-114
superconcentrators	
ON CONCENTRATION AND CONNECTION NETWORKS	TM-196
supervisors	
FINAL REPORT OF THE MULTICS KERNEL DESIGN PROJECT	TR-196
supervisory systems	
SUSPENSION OF PROCESSES IN A MULTIPROCESSING COMPUTER	TM-014
surveillance mechanisms	
MAKING COMPUTERS KEEP SECRETS	TR-115
SWALLOW	
MANAGEMENT OF OBJECT HISTORIES IN THE SWALLOW REPOSITORY	TR-243
swapping	
SUSPENSION OF PROCESSES IN A MULTIPROCESSING COMPUTER .	TM-014
switch-level simulation	
A SWITCH-LEVEL SIMULATION MODEL FOR INTEGRATED LOGIC .	TR-259
switching networks	
ON CONCENTRATION AND CONNECTION NETWORKS	TM-196
syllable	
PHRASE-STRUCTURE PARSING: A METHOD FOR TAKING ADVANTAGE .	TR-296
symbol manipulation	
THE REVISED MACLISP MANUAL	TR-295
symbolic integration	
SYMBOLIC INTEGRATION	TR-047
symbolic manipulation	
AN OVERVIEW OF OWL, A LANGUAGE FOR KNOWLEDGE .	TM-086
SYMBOLIC MATHEMATICAL LABORATORY	TR-036
SOME ASPECTS OF PATTERN RECOGNITION BY COMPUTER	TR-037
SYMBOLIC INTEGRATION	TR-047
symmetry codes	
SYMMETRY CODES AND THEIR INVARIANT SUBCODES	TM-044
synchronization primitives	
THE EVENT BASED LANGUAGE AND ITS MULTIPLE PROCESSOR .	TR-226
synchronization	
SEMAPHORE PRIMITIVES AND STARVATION-FREE MUTUAL	TM-158
A SEMANTICS OF SYNCHRONIZATION	TM-176
TIMEPAD - A PERFORMANCE IMPROVING SYNCHRONIZATION .	TM-177
SYNTHESIS OF SYNCHRONIZATION CODE FOR DATA ABSTRACTIONS	TR-203
NAMING AND SYNCHRONIZATION IN A DECENTRALIZED COMPUTER	TR-205
SYNCHRONIZATION MECHANISMS FOR MODULAR PROGRAMMING ..	TR-211
PARALLEL COMPUTATION· SYNCHRONIZATION, SCHEDULING, AND . .	TR-231
synchronous counters	
AUTONOMOUS, SYNCHRONOUS COUNTERS CONSTRUCTED ONLY OF .	TR-096
synchronous systems	
OPTIMIZING SYNCHRONOUS SYSTEMS	TM-215
syntax-based analysis	
SYNTAX-BASED ANALYTIC READING OF MUSICAL SCORES	TR-039
syntax-directed compilers	
DESIGN AND IMPLEMENTATION OF A TABLE-DRIVEN COMPILER	TR-042
syntax	
A FORMAL SYSTEM FOR DEFINING THE SYNTAX AND SEMANTICS .	TR-060
synthesis	
SYNTHESIS OF SYNCHRONIZATION CODE FOR DATA ABSTRACTIONS	TR-203
system design	
USE OF HIGH LEVEL LANGUAGES FOR SYSTEMS PROGRAMMING	TM-013
system implementation	
USE OF HIGH LEVEL LANGUAGES FOR SYSTEMS PROGRAMMING	TM-013
system initialization	
A SIMPLE AND FLEXIBLE SYSTEM INITIALIZATION MECHANISM	TR-180
system modeling	
USE OF HIGH LEVEL LANGUAGES FOR SYSTEMS PROGRAMMING	TM-013
system performance analysis	
USE OF HIGH LEVEL LANGUAGES FOR SYSTEMS PROGRAMMING	TM-013
system response time	
PERFORMANCE EVALUATION OF MULTIPROGRAMMED TIME-SHARED .	TR-103
systems analysis	
ANALYSIS AND SPECIFICATION OF OFFICE PROCEDURES	TR-275
OFFICE ANALYSIS METHODOLOGY AND CASE STUDIES	TR-289
systems programming	
USE OF HIGH LEVEL LANGUAGES FOR SYSTEMS PROGRAMMING	TM-013
systolic arrays	
WAFER-SCALE INTEGRATION OF SYSTOLIC ARRAYS	TM-236
systolic systems	
OPTIMIZING SYNCHRONOUS SYSTEMS	TM-215
table-driven compilers	
DESIGN AND IMPLEMENTATION OF A TABLE-DRIVEN COMPILER .	TR-042
tableau method	
DETERMINISTIC PROPOSITIONAL DYNAMIC LOGIC FINITE ..	TM-190
tableaux	
ALGEBRAIC DEPENDENCIES	TM-193
tail-recursion	
REFLECTION AND SEMANTICS IN A PROCEDURAL LANGUAGE	TR-272
tape complexity	
A SPACE BOUND FOR ONE-TAPE MULTIDIMENSIONAL TURING .	TM-145
SIMULATIONS AMONG MULTIDIMENSIONAL TURING MACHINES	TR-242
task system with resources	
CRITICAL PATH SCHEDULING OF TASK SYSTEMS WITH RESOURCE	TM-161
SCEDULING TASK SYSTEMS WITH RESOURCES	TR-236
tasking	
IMPLEMENTING INTERNET REMOTE LOGIN ON A PERSONAL	TM-233
technological incomprehensibility	
INCOMPREHENSIBLE COMPUTER SYSTEMS. KNOWLEDGE WITHOUT	TR-227
technological systems	
INCOMPREHENSIBLE COMPUTER SYSTEMS. KNOWLEDGE WITHOUT .	TR-227
telecommunications	
DATA COMMUNICATIONS VIA CABLE TELEVISION NETWORKS. .	TR-273
teleconferencing	
COOPERATIVE OFFICE WORK, TELECONFERENCING AND CALENDAR . .	TM-218
Telex	
A TELEX GATEWAY FOR THE INTERNET	TM-222
template and algebraic dependencies	
ALGEBRAIC DEPENDENCIES	TM-193
temporal knowledge	
MECHANIZATION OF TEMPORAL KNOWLEDGE	TR-155
tensor analysis	
COMPUTER PROGRAMS FOR RESEARCH IN GRAVITATION AND . .	TM-167
term rewriting systems	
RECURSIVE DECOMPOSITION ORDERING AND MULTISET ORDERINGS	TM-219
DERIVED PAIRS, OVERLAP CLOSURES, AND REWRITE DOMINOES.	TR-268
AUTOMATIC SYNTHESIS OF IMPLEMENTATIONS FOR ABSTRACT .	TR-276
terminal devices	
PROGRAM STRUCTURE IN A MULTI-ACCESS COMPUTER	TR-011

two-phase locking	
CONCURRENCY CONTROL FOR RESILIENT NESTED TRANSACTIONS	TR-285
type checking	
SEMANTICS OF DATA STRUCTURES AND REFERENCES	TR-134
type declarations	
LAMBDA CALCULUS MODELS OF PROGRAMMING LANGUAGES	TR-057
type extension	
USING TYPE EXTENSION TO ORGANIZE VIRTUAL MEMORY ...	TR-167
FINAL REPORT OF THE MULTICS KERNEL DESIGN PROJECT	TR-196
typed task systems	
BOUNDS ON THE SCHEDULING OF TYPED TASK SYSTEMS	TM-111
undecidability	
THE IMPLICATION PROBLEM FOR FUNCTIONAL AND INCLUSION ...	TM-235
understandability	
A PRINCIPLED DESIGN FOR AN INTEGRATED COMPUTATIONAL ...	TM-223
understanding office work	
EVALUATION OF AN OFFICE ANALYSIS METHODOLOGY	TM-239
AN OFFICE ANALYSIS AND DIAGNOSIS METHODOLOGY	TR-290
uniform termination	
DERIVED PAIRS, OVERLAP CLOSURES, AND REWRITE DOMINOES: ...	TR-268
universal computers	
INFORMATION PROCESSING AND TRANSMISSION IN CELLULAR ...	TR-081
unrelated processors	
ALGORITHMS FOR SCHEDULING TASKS ON UNRELATED PROCESSORS	TM-137
unreliable processes	
THE MUTUAL EXCLUSION PROBLEM FOR UNRELIABLE PROCESSES	TM-084
user models	
A PRINCIPLED DESIGN FOR AN INTEGRATED COMPUTATIONAL ...	TM-223
utilities	
TRAFFIC CONTROL IN A MULTIPLEXED COMPUTER	TR-030
validity problem	
COMPUTABILITY AND COMPLETENESS IN LOGICS OF PROGRAMS	TM-097
VAL	
AN INTERMEDIATE FORM FOR DATA FLOW PROGRAMS	TM-143
VAL--A VALUE-ORIENTED ALGORITHMIC LANGUAGE, PRELIMINARY ...	TR-218
APPLICATION OF DATA FLOW ARCHITECTURE TO COMPUTER MUSIC ...	TR-257
HIGH LEVEL VAL CONSTRUCTS IN A STATIC DATA FLOW MACHINE	TR-262
AN IMPLEMENTATION SCHEME FOR ARRAY OPERATIONS IN STATIC ...	TR-280
vector addition	
DECISION PROBLEMS FOR PETRI NETS AND VECTOR ADDITION ...	TM-059
SEMILINEAR SETS AND APPLICATIONS	TR-183
vector algebra	
CONSTRUCTION HEURISTICS FOR GEOMETRY AND A VECTOR ...	TM-028
vector replacement system	
AN EFFECTIVE REPRESENTATION OF THE REACHABILITY SET OF ...	TM-188
PERSISTENCE OF VECTOR REPLACEMENT SYSTEMS IS DECIDABLE	TM-189
THE COMPLEXITY OF THE WORD PROBLEMS FOR COMMUTATIVE ...	TM-199
verification	
REPORT ON THE WORKSHOP ON DATA FLOW COMPUTER AND ...	TM-092
VERIFICATION OF PROGRAMS OPERATING ON STRUCTURED DATA	TR-124
FINAL REPORT OF THE MULTICS KERNEL DESIGN PROJECT	TR-196
ABSTRACT MODEL SPECIFICATIONS FOR DATA ABSTRACTIONS	TR-221
AUTOMATIC VERIFICATION OF SERIALIZES	TR-229
very high level languages	
DATA DRIVEN LOOPS	TR-244
video terminals	
THEORY AND PRACTICE OF TEXT EDITORS OR A COOKBOOK FOR ...	TM-165
view complement	
TRANSLATING UPDATES OF RELATIONAL DATA BASE VIEWS	TR-284
view update	
TRANSLATING UPDATES OF RELATIONAL DATA BASE VIEWS	TR-284
viewpoints	
SUPPORTING ORGANIZATIONAL PROBLEM SOLVING WITH A ...	TM-224
virtual memory	
FILE MANAGEMENT AND RELATED TOPICS	TM-012
HARDWARE ESTIMATION OF A PROCESS' PRIMARY MEMORY ...	TM-081
DESIGN STRATEGIES FOR FILE SYSTEMS	TR-078
PERFORMANCE EVALUATION OF MULTIPROGRAMMED TIME-SHARED ...	TR-103
AN INPUT/OUTPUT ARCHITECTURE FOR VIRTUAL MEMORY ...	TR-117
AN EXPERIMENTAL ANALYSIS OF PROGRAM REFERENCE PATTERNS ...	TR-127
PROGRAM RESTRUCTURING FOR VIRTUAL MEMORY SYSTEMS	TR-148
USING TYPE EXTENSION TO ORGANIZE VIRTUAL MEMORY ...	TR-167
A CASE STUDY OF INTERMODULE DEPENDENCIES IN A VIRTUAL ...	TR-174
A LAYERED VIRTUAL MEMORY MANAGER	TR-177
vision	
COMPUTER RECOGNITION OF THREE-DIMENSIONAL OBJECTS IN A ...	TR-059
COMPUTER RECOGNITION OF PRISMATIC SOLIDS	TR-073
LEARNING STRUCTURAL DESCRIPTION FROM EXAMPLES	TR-076
SHAPE FROM SHADING: A METHOD FOR OBTAINING THE SHAPE OF ...	TR-079
COMPUTER ANALYSIS OF VISUAL PROPERTIES OF CURVED OBJECTS	TR-082
VLSI	
OPTIMAL PLACEMENT FOR RIVER ROUTING	TM-206
OPTIMIZING SYNCHRONOUS SYSTEMS	TM-215
LAYOUTS FOR THE SHUFFLE-EXCHANGE GRAPH BASED ON THE ...	TM-221
NEW LOWER BOUND TECHNIQUES FOR VLSI	TM-227
AN ASYMPTOTICALLY OPTIMAL LAYOUT FOR THE ...	TM-231
WAFER-SCALE INTEGRATION OF SYSTOLIC ARRAYS	TM-236
PLANAR EMBEDDING OF PLANAR GRAPHS	TM-237
AN APPROXIMATION ALGORITHM FOR MANHATTAN ROUTING	TM-238
AUTOMATIC TEST, CONFIGURATION AND REPAIR OF CELLULAR ...	TR-151
ARTWORK ANALYSIS TOOLS FOR VLSI CIRCUITS	TR-239
ALGORITHMS FOR INTEGRATED CIRCUIT LAYOUT: AN ANALYTIC ...	TR-248
A DESIGN METHODOLOGY FOR SELF-TIMED SYSTEMS	TR-258
A SWITCH-LEVEL SIMULATION MODEL FOR INTEGRATED LOGIC ...	TR-259
LAYOUTS FOR THE SHUFFLE-EXCHANGE GRAPH AND LOWER BOUND ...	TR-274
THE IMPACT OF LAYER ASSIGNMENT METHODS ON LAYOUT ...	TR-291
voltage waveform	
APPLICATION OF DATA FLOW ARCHITECTURE TO COMPUTER MUSIC ...	TR-257
wafer-scale integration	
WAFER-SCALE INTEGRATION OF SYSTOLIC ARRAYS	TM-236
weakest preconditions	
NONDETERMINISM IN LOGICS OF PROGRAMS	TM-098
LOGICS OF PROGRAMS: AXIOMATICS AND DESCRIPTIVE POWER	TR-200
well-founded ordering	
RECURSIVE DECOMPOSITION ORDERING AND MULTISET ORDERINGS	TM-219
well-structured programs	
THE PROPOSITIONAL DYNAMIC LOGIC OF DETERMINISTIC, ...	TM-198
wire length	
NEW LOWER BOUND TECHNIQUES FOR VLSI	TM-227
WAFER-SCALE INTEGRATION OF SYSTOLIC ARRAYS	TM-236
word problem	
THE COMPLEXITY OF THE WORD PROBLEMS FOR COMMUTATIVE ...	TM-199
working set	
HARDWARE ESTIMATION OF A PROCESS' PRIMARY MEMORY ...	TM-081
workshop report	
REPORT ON THE WORKSHOP ON DATA FLOW COMPUTER AND ...	TM-092
REPORT ON THE WORKSHOP ON SELF-TIMED SYSTEMS	TM-166
workstations	
SUPPORTING ORGANIZATIONAL PROBLEM SOLVING WITH A ...	TM-224
worst case finishing time	
ALGORITHMS FOR SCHEDULING TASKS ON UNRELATED PROCESSORS	TM-137
worst case performance	
ON THE WORST-CASE OF BEHAVIOR OF STRING-SEARCHING ...	TM-071
AN ANALYSIS OF PREEMPTIVE MULTIPROCESSOR JOB SCHEDULING	TM-110
BOUNDS ON THE SCHEDULING OF TYPED TASK SYSTEMS	TM-111
EFFICIENT SCHEDULING OF TASKS WITHOUT FULL USE OF ..	TM-122
worst-case performance	
ON THE WORST-CASE OF BEHAVIOR OF STRING-SEARCHING ...	TM-071

ATTACH UPDATE HERE

ATTACH UPDATE HERE

ATTACH UPDATE HERE

ATTACH UPDATE HERE

ATTACH UPDATE HERE

ATTACH UPDATE HERE

September 1983

PLEASE NOTE THE FOLLOWING REVISIONS TO THE LCS BIBLIOGRAPHY:

TR-295 -- Please note price adjustment.
Pitman, K.M.
THE REVISED MACLISP MANUAL
Pages: 325 June 1983 $8.90

TR-296 -- Please note price adjustment and title.
Church, K.W.
PHRASE-STRUCTURE PARSING: A METHOD FOR TAKING ADVANTAGE OF ALLOPHONIC CONSTRAINTS
Pages: 305 Ph.D. dissertation/January 1983 $13.50

TR-297 -- Please note that this publication is not yet available.
Mok, A.K.
FUNDAMENTAL DESIGN PROBLEMS OF DISTRIBUTED SYSTEMS FOR THE HARD-REAL-TIME ENVIRONMENT
Pages: 150 Ph.D. dissertation/June 1983

TR-298 -- Please note price adjustment.
Krugler, K.
VIDEO GAMES AND COMPUTER AIDED INSTRUCTION
Pages: 45 August 1983 $6.30

TR-299 -- New Publication
Wing, J.M.
A TWO-TIERED APPROACH TO SPECIFYING PROGRAMS
Pages: 164 Ph.D. dissertation/May 1983 $8.95
Keywords: formal specifications, program design, specification languages, specification analysis, algebraic specifications, abstract data types, programming methodology.
Abstract: Current research in specifications is beginning to emphasize the practical use of formal specifications in program design. This thesis presents a specification approach, a specification language that supports that approach, and some ways to evaluate specifications written in that language.

The two tiered approach separates the specification of underlying abstractions from the specification of state transformations. In this approach, state transformations and target programming language dependencies are isolated into an interface language component. All interface specifications are built upon shared language specifications that describe the underlying abstractions. This thesis presents an interface specification language for the CLU programming language and presumes the use of the Larch shared language.

This thesis also suggests a number of kinds of analyses that one might want to perform on two-tiered specifications. These are related to the consistency, completeness, and strength of specifications, and are all presented in terms of the theories associated with specifications.

TR-300 -- New Publication
Cooper, G.H.
AN ARGUMENT FOR SOFT LAYERING OF PROTOCOLS
Pages: 117 M.S. Thesis/May 1983 $7.75
Keywords: computer networks, layered protocols, soft layering
Abstract: This thesis is about the efficiency of protocol layering. It examines the technique of protocol layering in an abstract way and finds two major sources of inefficiency in protocol implementations which are caused by the imposition on them of a layered structure. The conventional approach to making layered protocol implementations run efficiently -- for avoiding the sources of inefficiency discussed herein -- are all independent of the protocol specification, and thus all decrease the value of the protocol specification

as a guide for implementing protocols.

In this thesis, we introduce a new means of avoiding the problems of layered protocol implementations which operates within the domain of the protocol specification. We allow an increase in the flow of state information between the layers of a layered protocol implementation in a very controlled manner, so as to decrease the modularity of the protocol architecture as little as possible. The increased flow of information is specified in the protocol specification as a model of all the layered protocols that use the protocol being specified, called the "usage model." Since our approach decreases the rigidity of the layered structure without entirely eliminating it, we coin the term "Soft Layering" for the approach.

Miscellaneous Publications

MACSYMA REFERENCE MANUAL -- **Version 10**
Three Volumes January 1983 $22.50
Abstract: MACSYMA (Project MAC's SYmbolic MAnipulation System) is a large computer programming system written in LISP used for performing symbolic as well as numerical mathematical manipulations. With MACSYMA the user can differentiate, integrate, take limits, solve systems of linear or polynomial equations, factor polynomials, expand functions in Laurent or Taylor series, solve differential equations (using direct or transform methods), compute Poisson series, plot curves, and manipulate matrices and tensors. MACSYMA has a language similar to ALGOL-60 to permit the user to write his own programs for transforming symbolic expressions.

This manual is intended to be a complete reference for the principal features of MACSYMA as of January 1983. It is not meant as a tutorial nor does it discuss all of the issues involved in the efficient manipulation of algebraic expressions.

BARRIERS TO EQUALITY IN ACADEMIA: WOMEN IN COMPUTER SCIENCE AT MIT
Pages: 44 February 1983 5.00
Abstract: This report describes aspects of the MIT computer science environment that tend to hinder the professional and social development of some of the female graduate students and research staff. In an environment that, for both men and women is challenging, competitive, and difficult, some women encounter additional problems that may unfairly limit their academic, professional, and personal growth. The goal of this report, written by female graduate students and research staff in a predominantly male environment, is to heighten the awareness of these problems and their effects among members of the technical community both within and outside MIT.

MIT/LCS PROGRESS REPORT 18
Pages: 379 July 1980-June 1981 Free

MIT/LCS BROCHURE
Pages: 28 Fall 1982/Spring 1983 Free

Updates to the LCS Bibliography will be revised and supplied regularly.

Note that $0.60 is added to the price of each publication to cover postage and should therefore be deducted for in-house sales.

ORDERING INFORMATION

Please use the following format when ordering publications:

TM/TR #	Author	Quantity	Price	Total
				$

Requestor's Name: ______

Address: ______

Prepayment is required on all orders. Please enclose a check, money order or purchase order with your request. (Prices include surface postage.) Send to:

MIT Lab for Computer Science
Publications Unit
545 Technology Sq.
Cambridge, MA 02139

NOTE: Publications which list an "AD" or "PB" number may also be obtained from the National Technical Information Service, U.S. Dept. of Commerce, 5285 Port Royal, Springfield, VA 22150 (703) 487-4650.