DESCRIPTION OF
GRADUATE COURSES IN COMPUTER SCIENCE
1972-73

SAUL AMAREL, CHAIRMAN
DEPARTMENT OF COMPUTER SCIENCE

NEW BRUNSWICK, N.J.
JULY, 1972
INTRODUCTION

Here is a collection of comprehensive descriptions of the graduate courses in computer science that are being offered in the academic year 1971-72 by the Department of Computer Science.

Each course description specifies the academic objectives of the course (also, for what students it is intended and at what stage of their graduate studies); the preparation expected of a student taking the course (course prerequisites and other required knowledge or experience); the work expected of a student in the course (assignments, examinations, computer projects; an outline of the course; and references (textbooks, key readings, relevant papers) that provide a further basis for identifying the subject matter and level of the course.

The primary purpose of these descriptions is to assist students in the job of planning their academic program. On basis of the present course descriptions, and the departmental brochure "Graduate Program in Computer Science, 1971-72", of July 1971, a student (possibly in consultation with members of the DCS faculty) should be able to develop his overall study plan in considerable detail. Of course, each student must discuss his plan with (and receive the approval of) his faculty advisor, before the plan goes into effect.

Important: Prerequisites are to be taken as strong guidelines that should help a student to decide whether to register in a specific course at a given time in the course of his studies. Normally, students taking a course should satisfy the course prerequisites. However, if a student is unclear about the prerequisites of a course or if he has doubts about his qualifications for the course, he should consult the instructor in charge. Special cases will be considered individually by the instructor.

The overall structure of our graduate computer science courses is summarized diagrammatically in Figure 1; boxes stand for courses, and lines denote relations between courses and their prerequisites. Planning for a program of courses may be carried out in a straightforward manner with the help of this diagram. Note that odd numbers designate Fall courses, and even numbers Spring courses. In cases where a Fall course is repeated in the Spring, an S is added after the course number (e.g. 503S is given in the Spring).
Notes for Figure 1:

- Indicates a required previous course.
- - Indicates a recommended previous course.
& Means that all entering paths are required or recommended.
V Means that any of the entering paths is acceptable.

Dark boxes indicate required courses; 503, 507 or 508, 510 and 513 are requirements for all graduate degree candidates; also, 611 or 612 is a requirement for doctoral candidates.

Meaning of Labels Appearing in Figure 1:

C\textsubscript{1} Computers and machine level programming (equivalent to 198:301-302 or 198:311-312).

C\textsubscript{2} High level language programming (e.g. PL/I, FORTRAN at the level of 198:211 or 198:212).

C\textsubscript{3} Non-numerical programming at level of 198:212 and logic & algebra at level 198:205.

M\textsubscript{1} Minimum of 4 terms of undergraduate mathematics.

M\textsubscript{2} Same as M\textsubscript{1}, including calculus and linear algebra (at least the equivalent of 198:105-106).

M\textsubscript{3} Same as M\textsubscript{2}, including, in addition, probability theory (at least equivalent to 198:105-106, 205, 206).

M\textsubscript{4} Modern algebra and combinatorials (at least equivalent to 198:205,206).

M\textsubscript{5} Linear algebra and matrix methods (at the level of 198:105-106 and 640:251).

M\textsubscript{6} Real and complex analysis (intermediate undergraduate level).

M\textsubscript{7} Algebraic topology (level of 640:451-452).

M\textsubscript{8} Any of the courses: Math 351-352, Math 354, or IE 410.

L\textsubscript{1} Intermediate logic (equivalent to Philosophy 332).

L\textsubscript{2} Mathematical logic (Philosophy 532).

G\textsubscript{1} Minimum of 3 graduate courses in computer science.

G\textsubscript{2} At least one of the courses CS 507, 528, 530, 539.

G\textsubscript{3} Minimum of 4 graduate courses in computer science.

G\textsubscript{4} Minimum of 6 graduate courses in computer science.

s Indicates special prerequisites that change (and will be announced) from year to year, depending on the main theme of the course.
Computer Science 503 and 503S  ADVANCED PROGRAMMING TECHNIQUES  (3 credits)

Instructors  I. Rabinowitz (Fall), G. Falk (Spring)

Schedule:  Fall Semester (01 M 6:15-9:00 - H11-122) Spring: M 6:15-9:00
(02 W3 F3 H11-120)

Objectives  Required of Computer Science degree candidates. This course
is designed to provide some of the tools of general utility
in programming, with major emphasis on the machine-oriented
aspects of these techniques. It concentrates on two areas,
Data Structures and Program Structures--how they are repre-
sented in present-day machines, and how they are manipulated
at the machine level.

Prerequisites  198:301-302 or 198:311-312, or equivalent programming experience,
including knowledge of machine-language programming for some
computer. If this equivalent is offered, consent of the in-
structor must be obtained.

Outline  I.  Data Structures

   a) Arrays with explicit storage mapping functions:
      vectors, matrices, triangles; recursive subscripting.

   b) Linked Structures: one-way, two-way, circular linked
      lists; methods of sequencing and searching; trees:
      creating, traversing, copying; stacks and queues.

   c) Applications: lexical analysis, finite-state grammars;
      symbol tables, linear search, binary search, binary trees;
      hash coding methods; sorting examples, interchange,
      merge, radix; storage allocation, garbage collection.

II.  Program Structures

   a) Static representation: fixed and variable parts, re-
      entrant programs.

   b) Procedures: Linkage and arguments, calls by value,
      reference, name, result; subroutines and co-routines.

   c) Block structure and recursion: Scope of names, stack
      implementation, activation records.

Expected Work  The course is primarily a lecture course with both paper home-
work assignments and programming exercises. The primary
programming language is MIXAL, defined in the book by Knuth
(see below).

References  Textbooks: Knuth, D.E., "Fundamental Algorithms," Addison-
Wesley, 1968.

Wegner, P., "Programming Languages, Information Structures,

(continued)
503 & 503S (continued)


Reference is made during the term to papers in CACM, JACM and other journals.

July 1972
Computer Science 504  ASSEMBLERS, LOADERS, AND MACROS  (3 credits)

Instructor  I. Rabinowitz

Schedule  Spring semester - 01  T 6:15-9:00

Objectives  This course presents the machine-oriented structure of assemblers and loaders, and discusses macro-processors from the following points of view: as definitional facilities for language extension, and as implementations of models of evaluative mechanisms. It is largely concerned with interpretations of those structures whose static properties have been shown in 503.

Prerequisites  CS 503 or equivalent and knowledge of some "higher-level language," e.g., PL/I, Fortran, or Algol. (equivalent to 198:211 or 198:212).

Outline  I. Assemblers and Loaders: Symbol Tables, one-pass and two-pass assemblers; assembly-time directives; relocatable and link-editing loaders.

II. Macros: Macro-processors over a ground language; independent macro-processors.

III. Evaluative mechanisms: Representation of programs by abstract syntax; evaluation mechanisms: macro-processors and abstract machines.

Expected Work  The course is primarily a lecture course with some paper homework and one term project, a macro-processor. It is recommended that the project be implemented in a higher-level language.


Computer Science 507 and 507S  INTRODUCTION TO SEQUENTIAL MACHINES  (3 credits)

Instructor  
E.J. Wilkens and A. Yasuhara (Fall), A. Yasuhara (Spring)

Schedule  
Fall Semester, 01 - Th. 6:15-9:00 Hil-122  Spring: Th. 3:15-6:00  
02 - M3 Th 2 Hil 120

Objectives  
This course or CS 508 are required of Computer Science degree candidates. Introduction to basic building blocks and techniques used in constructing digital computers. The concepts are also widely applicable in programming.

Prerequisites  
A minimum of four terms of undergraduate mathematics (the normal mathematical background of a science or engineering undergraduate major sufficient).

Outline  
In the first part of the course a finite state system is defined. A finite state system is a means by which a class of functions which map sequences into sequences can be defined. Such functions are of interest because they are the functions which are performed by much of the component hardware which makes up a computer and other digital devices. They also appear in analysis and translation of programming languages. Initially, many of the abstract properties of finite state systems are studied without much reference to their applications. Various equivalences, means of verifying the behavior and alternate representations such as regular expressions of finite state systems will be studied. The first five chapters of reference 1 are used as a text for this phase of the course.

The second phase is devoted to considerations of how to actually construct sequential machines which exhibit the functional behavior defined by a finite state system. The construction is carried out in terms of idealized elements with very simple behavior. These include 'logic gates' and 'flip-flops'. This does not include detailed consideration of design at the transistor level which would require more physics and mathematics background. The mathematics appropriate for this level of design is discrete mathematics including the development of boolean algebra--and this is studied in the course. Applications of boolean algebra in the design of sequential circuits is considered, also various techniques for constructing efficient sequential machines. This part of the course uses the first four chapters of reference 2.

Expected Work  
Homework assignments; midterm and final examinations.

References  


July 1972
Computer Science 508 and 508S  MACHINE ORGANIZATION  (3 credits)

Instructor  
S.Y. Levy

Schedule  
Fall: 01  T 6:15-9:00 Hil 120  Spring: 01  T. 3:15-6:00
02  W. 3:15-6:00

Objectives  
This is a required course for graduate students in Computer Science, as an alternate to CS 507. The course will introduce fundamental concepts in computing machine organization and machine design.

Prerequisites  
Familiarity with computing machines at the assembly language level (198:301-302 or 198:311-312 or equivalent.) Knowledge of logical design and switching theory will be useful, but not essential.

Outline  
Principles of computer organization. Typical machine architectures. Selection of instruction repertoire, number representation and arithmetic-operations, non-numerical operations, instruction scheduling, design of data-flow and controls, elementary controls and concept of microprogramming, building blocks for system design, and elementary concepts in logical design. Memory organization, and Input/Output systems. Design problems and design methods will be discussed.

Expected Work  
Homework assignments, mid-term and final exam. Interested students will be encouraged to undertake project work in the area of computer subsystem simulation.

References  
Reference books and articles of interest will be discussed in the class. Lecture notes will be distributed to the students.

A. Newell, G. Bell, "Computer Structures"

W. Buckholtz, et al, "Planning a Computer System"

Special Issue of IBM J. of R & D: on 360 Model 91,

Papers of interest in the technical literature.

J.B. Dennis, S.S. Patil, "Notes on Computation Structures"

July 1972
Computer Science 510      NUMERICAL ANALYSIS (3 credits)

Instructors        F. G. Fender (Spring)  R. Vichnevetsky (Fall)

Schedule          Fall: 01 W 6:15-9:00  Hil-122  Spring: 01 W 3:15-6:00
                  02 W 6:15-9:00

Objectives        Introduction to the methods used in the solution of
                   numerical problems by computers.  Applications are
                   emphasized.

Prerequisites      Programming facility with a high level language, such as
                   PL/I or FORTRAN IV (at the level of 198:211 or 198:212):
                   a minimum of four terms of undergraduate mathematics, including
                   calculus and linear algebra.

Outline            Each of the following topics will be briefly outlined, and
                   followed by an analysis of its properties in terms of stability,
                   error, convergence, etc.:
                   Solution of transcendental equations by iterative methods,
                   systems of linear equations (matrix inversion, etc.).  Approxima-
                   tion and computer generation of functions.  Interpolation, dif-
                   ferentiation and quadrature.  Solution of ordinary differential
                   equations.
                   Optimum searching methods.

Expected Work      The student may expect a computer problem and a theoretical
                   (analysis) problem every two weeks, two take-home examinations
                   and a (take-home?) final.  The computer language used will be
                   FORTRAN or PL/I.

References         Textbook:  Ralston, "A First Course in Numerical Analysis,"
                   Other books:  Hamming, "Introduction to Applied Numerical
                   Isaacson and Keller, "Analysis of Numerical Methods,"

July 1972
Computer Science 513 and 513S  NON-NUMERICAL ALGORITHMS  (3 credits)

**Instructors**  
M.C. Pauli and B. Bruce (Fall), B. Bruce (Spring)

**Schedule**  
Fall Semester  : 01 T 3:15-6:00 Hil-122  
Spring: 01 T 3:15-6:00  
02 M4  
T4 - Hil-122

**Objectives**  
A required course for Computer Science degree candidates. Discussion of representative algorithms and data structures encountered in non-numerical applications.

**Prerequisites**  
Programming facility with some high level language, e.g., PL/1, (equivalent to 198:211 or 198:212); undergraduate courses in modern algebra and combinatorials (equivalent to 198:205-206).

**Outline**  
Problems from the following areas will be discussed: symbol manipulation in mathematics and in language studies; heuristic problem solving in theorem proving, game playing, and other complex situations; problems in graphics and simulation.

Emphasis will be placed on developing the basic concepts and techniques which recur in these problem areas. This includes the construction of algorithms for manipulating data-structures, such as trees and more general graphs; the use of recursive definitions; the formulation of procedures for pattern matching, problem reduction and search.

Comparison and evaluation of different approaches including the relative value of different programming languages will be discussed.

**Expected Work**  
The course is primarily a lecture course with homework assignments--including the design and programming of algorithms in some high level language (e.g., PL/1); final examination or term project.

**References**  


July 1972
Computer Science 515  PROGRAMMING LANGUAGES AND COMPILERS I  (3 credits)

Instructors  M. C. Paull, I. N. Rabinowitz

Schedule  Fall Semester, 01 Th 3:15-6:00 Hi1-122

Objectives  This introduces the concepts and techniques used in the description of programming languages and in the construction of their translators.

Prerequisites  Knowledge of computers and machine-level programming equivalent to 198:301-302 (or 198:311-312); experience with high-level programming languages (equivalent to 198:211 or 198:212).


Expected Work  A number of programming problems will be assigned, to be done in some higher-level language.


July 1972
Computer Science 516  PROGRAMMING LANGUAGES AND COMPILERS - II. (3 credits)

Instructors  M. C. Paull, I. Rabinowitz

Schedule  Spring Semester, 01  Th 3:15-6:00

Objectives  This course is designed to cover some advanced aspects of compiler construction.

Prerequisites  CS 503 and CS 515.

Outline  A more extended and detailed coverage of compilation problems where CS 515 has laid the groundwork. The main effort of the course will be directed toward the study of the detailed techniques involved in the construction of a practical compiler for complex algebraic languages, for translation to a machine language. The course will cover questions about choice of a language description, parsing methods, code generation and optimization, structure of a "compiler-compiler", storage allocation methods, and other topics.

Expected Work  The construction of a complete compiler of a complexity approaching that of a compiler for the BASIC language.

References  Textbooks:


Reference will be made to original papers in various journals.

July 1972
Computer Science 517  SYSTEMS MODELING  (3 credits)

Instructor:  K. R. Kaplan

Schedule:  Fall semester, 01 M 6:15-9:00 Hil-124

Objectives:  To develop an applied mathematics orientation toward the modeling of complex systems, and to permit easier access to advanced literature in computer system evaluation. For students interested in analytic modeling and computer simulation.

Prerequisites:  A minimum of four terms of undergraduate mathematics; a good background in calculus; inclination toward mathematical modeling. A knowledge of computers and programming equivalent to 198:301-302 or 198:311-312.

Outline:  The lectures develop from fundamental concepts in probability theory, statistics, random processes, queueing theory and markov chains. Whenever possible, the tutorial problems are augmented by application of the material to the modeling and evaluation of computer subsystems and configurations, e.g., scatter-table "hashing," time-sharing systems, processor-memory performance, multiprocessor interference, memory hierarchies, etc.

Expected Work:  Several (3-4) homework assignments of 3-5 problems each; in-class midterm, and a take-home final exam.


July 1972
Computer Science 519  OPERATING SYSTEMS  (3 credits)

Instructor  W. Easton

Schedule  Fall semester, 01 T 6:15-9:00 Hil-252

Objectives  For students with some experience or knowledge in systems programming who are interested in the design and construction of operating systems.

Prerequisites  One year of systems programming or CS 503-504, or equivalent. Programming competence with a higher level language (equivalent to 198:211 or 198:212). PL/I will be used as a 'spoken language' in the classroom; however, a detailed knowledge of PL/I is not required. Students should be familiar with assembly language programming on some machine (equivalent to 198:301-302 or 198:311-312).

Outline:  Processes, inter-process communication, problems of multiprogramming and multiprocessing, organization and queuing of mass storage, file structures, paging and segmentation.

Expected Work  Students will be asked to write a portion of a multiprogramming operating system "to run in a simulated environment" under the computer system available to them, using a higher level language. In the past, students have used PL/I, FORTRAN, COBOL, ALGOL, and SNOBOL 4.


July 1972
Computer Science 521  COMPUTATIONAL METHODS FOR OPTIMIZATION PROBLEMS  
(3 credits)

Instructor  M. D. Grigoriadis

Schedule  Fall Semester - W 6:15-9:00  MU 203

Objectives  This course introduces concepts, algorithms and computational methods used in the solution of large scale linear optimization problems. Some algorithms for nonlinear problems will also be reviewed. The information processing problems involved and applications in the decision, physical and computer sciences will be presented.

The course is intended for computer science students, and also for students in mathematics, engineering, urban planning and economics who are interested in computer approaches to large optimization problems.

Prerequisites  Either of the following courses are acceptable prerequisites: CSS 510, or CS 513 with some knowledge of matrix algebra, or Math 351-352, or Math 354, or IE 410. Knowledge of a high level programming language such as PL/I or FORTRAN or consent of instructor.

Outline  Brief review of matrix and linear algebra, convex sets and functions. Review of the mathematical programming (MP) problem, Lagrange multipliers, optimality conditions and duality.

Primal and dual linear programming (LP) algorithms, degeneracy, scaling, error control and reinverson techniques. Design and organization of large LP codes, problem control language, matrix generation and report writing systems. Applications from the industrial and public sectors (e.g., models of air pollution).

Large structured LP problems. Decomposition and partitioning algorithms. Basis compactification schemes. Flow networks and mixed large scale models.


Expected Work  One final take-home examination. Group and individual assignments, some requiring computer processing (program development in PL/I or FORTRAN, and use of existing packages.)

References  Books (by Noble, Simonnard, Lasdon, Dantzig, Orchard-Hays, Berge & Gouila-Houri, Hadley) and research papers will be discussed in class.

July 1972
Computer Science 522 DEVELOPMENT AND ANALYSIS OF ALGORITHMS (3 credits)

Instructor
M. D. Grigoriadis

Schedule
Spring Semester, W 6:15 - 9:00 p.m.

Objectives
This course offers an introduction into a variety of discrete optimization problems, algorithms and computational methods used for their solution. More detailed studies on some of these problems emphasizing their computational complexity will be presented. Significant applications will be discussed. The course is intended for students in computer science, mathematics, engineering, urban planning and economics interested in computer approaches to large discrete optimization problems.

Prerequisites
Elementary knowledge of linear programming and basic concepts in graph theory and the following courses: CS 510 or CS 513, and Math 351-352 or Math 354 or IE 410 or CS 521 or consent of instructor. Knowledge of a high level programming language such as PL/I or FORTRAN or consent of instructor.

Outline

Integer and mixed integer programming. Approximate, branch and bound, and enumerative methods. Organization of search data. Special methods for set covering, facilities location, critical path networks with resource constraints, class scheduling, fixed charge problems.

Expected Work
One final take-home examination. Group and individual assignments, some requiring computer processing (program development in PL/I or FORTRAN, and possible use of existing packages.)

References
Books by Dantzig, Berge & Gouila-Houri, Hu, Knuth, Busacker and Saaty, and research papers will be discussed in class.

July 1972
Computer Science 523  COMPUTER GRAPHICS  (3 credits)

Instructor:  S. Baxendale

Schedule:  Fall semester, 01 T 4:45 - 7:30 Hil-254

Objectives:  Course is intended for both the student who requires a knowledge of computer graphics in his work and the student seeking an area of specialization in computer science.

Prerequisites:  Knowledge of linear algebra (e.g., 198:105-106) and matrix methods (e.g., 640:251); knowledge of computers and machine level programming equivalent to 198:301-302 or 198:311-312; facility with high level programming languages (equivalent to 198:211 or 198:212).


Expected Work:  Homework assignments; programming of plotter projects on drum plotter and CRT plotters, and reports on projects; visits to computer graphics installations; term paper.


July 1972
Computer Science 525, 526  ADVANCED NUMERICAL ANALYSIS  (3 credits each semester)

Instructor:  F. G. Fender

Schedule:  Fall semester,  01  W 3:15-6:00  CA A-3
            Spring semester,  01  M 6:15-9:00

Objectives:  For graduate students who wish a deeper experience in the art and the science of using a programmed digital computer in the numerical solution of mathematical problems.

Prerequisites:  CS 510 or equivalent; familiarity with Real and Complex Analysis (at the level of Whitaker and Watson "Modern Analysis," Cambridge U. Press). facility with a high level programming language such as PL/I or FORTRAN IV. Consent of instructor is required.

Outline:  Norms and well-posed computations; Numerical Solution of linear systems and operations with matrices; Eigenvalues and eigenvectors of matrices; Non-linear equations and iterative solutions; Roots of polynomials, Bairstow and Q-D methods; Introduction to approximations and interpolation.

(End of First Semester.)

Differentiation and integration of an experimental table of data; Solutions of ordinary differential equations; Solutions of partial differential equations--elliptic, hyperbolic and parabolic types, stability.

Expected Work:  The student will be given about ten computer projects a semester. He will also be given take-home tests and a final examination which will require computer processing. The computer languages used will be PL/I or FORTRAN IV.


July 1972
Computer Science 528 - THEORY OF FORMAL LANGUAGES (3 credits)

Instructor: A. Yasuhara

Schedule: Spring semester, 01 Th. t:16-9:00

Objectives: This course is intended to present the formal basis for many of the concepts and techniques used in the construction of programming language compilers.

Prerequisites: CS 515 and undergraduate courses in modern algebra and logic equivalent to 198:205, 206, or CS 507 and 198:301-302 (or 198:311-312 or equivalent) and some experience in higher level programming languages (e.g. 198:211 or 198:212)

Outline: The course will cover the definition and properties of phrase structure grammars with major emphasis on context-free and finite state grammars. These grammars provide means for defining programming (and other) languages. Various equivalences between grammars will be developed, including Chomsky and Greibach Normal Form. The limitations of the various grammars as to the kinds of languages they can define will be covered. Methods for analyzing languages defined by such grammars will be formally developed. These methods are given in the form of automata. We will study finite state, push-down, and Turing automata which are in turn adequate for analyzing more and more inclusive languages.

Expected Work: For the most part the course will follow the material in the reference, although some topics too recently developed to be included there, such as optimizing of compiler output code, will also be covered. No extensive programming will be required in this course. Regular homework assignments, largely drawn from the text, will be required. There will be a midterm and final exam.


July 1972
Computer Science 529 - NATURAL LANGUAGE PROCESSING AND QUESTION ANSWERING BY COMPUTER (3 credits)

Instructor
B. Bruce

Schedule
Fall Semester, 01 W 3:15-6:00 TBA

Objectives
This course examines various approaches to processing natural language by computer. Special emphasis is placed on deductive systems for question answering.

The course is intended for computer science students, and also for students in other areas (linguistics, philosophy, psychology, anthropology, library science, education) who are interested in computer-based methods for the manipulation and study of natural language.

Prerequisites
CS 513; or programming facility in at least one high level programming language (preferably experience in LISP programming), knowledge of elementary algebra, the level of 198.205, and experience in mathematics and algorithms equivalent to 198.202. Consent of instructor.

Outlining
The first half of the course surveys various approaches to natural language processing. Computer programs written in LISP or prolog automatic translation, for fact retrieval by natural language commands, for use in diagnosis of mental problems will be applied. Data processing will be briefly discussed. Included also will be some discussion of several semantic theories, such as Katz-humor and transformational grammars, which are relevant to computer science.

Intermediate (intermediate) course in semantics: a minimum prerequisite.

Expected Work
Selected readings from texts and research papers, a term project consisting of a complete, but small scale, question answering program, or a portion of such a program (such as a parser) developed in greater detail.

References
Computer Science 535 - PATTERN RECOGNITION. THEORY & APPLICATIONS (3 credits)

Instructor C. Kulikowski

Schedule Fall Semester, 01 W 6:15-9:00 Hil-252

Objectives The principal purpose of this course is to introduce the student to the problems of pattern recognition through a comparative presentation of methodology and practical examples. The course is intended for computer science students with an applied mathematics orientation, and also for students in other programs (electrical engineering, statistics, mathematics psychology) who are interested in this area of research.

Prerequisites Undergraduate courses in calculus and linear algebra (minimum level of 198:105-106; preferred level 640:311-312, 351-352), and in probability/random variables (minimum level of 198:205-206, preferred level 640:477-478). The courses CS 507 and CS 513 are recommended, but not required.

Outline Cognition and recognition: pattern recognition as an inductive process. Feature extraction methods. Similarity measures and probability. Statistical classification: recognition as a decision problem; optimum Bayes' rules; optimum minimax procedures; maximum likelihood decisions. Linear and nonlinear discriminants. Structure of data; principal components and factor analysis. Adaptive pattern recognition: linear training; perceptrons; potential functions and stochastic approximation. Linguistic methods in picture processing: generative procedures and transformational grammars. Sequential methods in classification. Applications in visual and character recognition; speech recognition; automated medical diagnosis; chromosome identification; analysis of EKG and EEG.

Expected Work The student is required to undertake a project or write a term paper in this course. A final examination covering the main concepts of the course is also given.


July 1972
Computer Science 537S  OPTIMIZATION THEORY AND COMPUTATIONAL METHODS (3 credits)

Instructor
R. Vichnevetsky

Schedule
Spring Semester  01  T  3:15-6:00

Objectives
An introduction to the theory of optimization in parameter and function space. Emphasis is placed on the principles and techniques used in the implementation of computer programs for applications, with illustrations.

The course is intended for computer science students with an interest in advanced numerical applications, and for students in other areas (engineering, mathematics) who are interested in theoretical and computational aspects of optimization processes.

Prerequisites
A reasonable background in differential equations, matrix theory and numerical analysis up to numerical integration (at least equivalent to CS 510 with some knowledge of matrix algebra); or consent of instructor.

Outline
A. Optimization in parameter space:
Ordinary extremum, equality constraints and Lagrange multipliers, geometrical interpretations, inequality constraint and Kuhn and Tucker's theorem.

B. Computational methods in parameter optimization:
Gradient methods, variable metric methods, pattern search, random search.

C. Optimization in function space:
Dynamic systems, perturbation equations, adjoint equations, Green functions, necessary and sufficient conditions for an optimum, Pontryagin's maximum principle, Hamilton-Jacobi-Bellman equation, geometrical interpretation and Huyghen's principle, classical variational calculus formulation, relation to classical mechanics, dynamic programming.

D. Computational methods for optimization in function space:
General concepts; computing errors propagation, direct methods, indirect methods, gradient methods, Newton-Raphson method.

Expected Work
Homework assignments, selected readings, and a final examination.

References
A. Printed notes for most of the course material will be available.

B. Books:
Wilde, D.J. "Optimum Seeking Methods" Prentice-Hall, 1963
Gel'fand, I.M. & Fomin, S.V. "Calculus of Variations, (as above)
Bryson, A. & Ho, Y. "Optimal Control" Ginn & Blaisdell, 1969.

C. Selected articles from the technical literature.

July 1972
Computer Science 539 - THEORY OF COMPUTATION  (3 credits)

Instructor: R. Orgass

Schedule: Fall semester, 01 M 3:15-t:00 Hil-122

Objectives: This course is intended for advanced Master's Degree candidates and Doctoral Candidates. The objective is to introduce students to various topics in the theory of computation. The topics selected should be of interest to students whose main interest in programming languages and their processors. Knowledge of the subject matter of this course is essential for students whose main interest is in the theoretical aspects of computer science.

Prerequisites: Philosophy 532 "Mathematical Logic" (it will be assumed that students have a solid understanding of the material in Philosophy 532); the courses CS 503 and CS 507 or their equivalent; one of the courses CS 510 or CS 513, or equivalent. Students who have some doubt about their qualifications should meet with the instructor to discuss the course.

Outline: The course will begin with a study of important and applicable results in the theory of computable functions, i.e., Turing machines and recursion equations. The emphasis in this study will be on decidability and undecidability results which are relevant to problems in computer science. Following this, there will be a study of mathematical models of computing machines and programming languages. Several models will be discussed and compared. Finally, there will be a discussion of the problem of verifying the correctness of programs and of the equivalence of programs.

In particular the work of Floyd, Manna, Scott, and Orgass will be discussed.

Expected Work: Selected readings from standard textbooks and research papers; assignments of review and critique papers.

References: The following papers are typical of those which will be studied in the course.


July 1972
Computer Science 542 - PROBLEMS IN SIMULATION (3 credits)

Instructor: K. R. Kaplan

Schedule: Spring semester, 01 M 6:15-9:00

Objectives: To develop familiarity and skills in computer simulation using the languages, GPSS and SIMSCRIPT. For students needful of a general purpose simulation capability for graduate projects or industrial applications.

Prerequisites: CS 517 or equivalent; one of the courses CS 510 or CS 513 or equivalent.

Outline: The course consists of formal lectures in GPSS and SIMSCRIPT, interspersed with workshop sessions for monitoring the progress of student projects.

Expected Work: Students form groups of from 1-3 members. Each group prepares a project statement describing a simulation experiment. Projects from the computer science area are preferred, but other problem areas of sufficient depth and maturity are acceptable. The project groups model, program, experiment with, and validate the system they have chosen. Example study: Programming and evaluating a simulation of a simple model of a multi-programming system. Final reports are required of each group.


Several research papers in simulation will be distributed in the course.

July 1972
Computer Science 544 - PROBLEMS IN COMPUTER GRAPHICS (3 credits)

**Instructor:** S. Baxendale

**Schedule:** Spring semester, 01 T 4:45-7:30

**Objectives:** Course intended for students interested in advanced work in computer graphics. It provides a good opportunity of project work for doctoral candidates.

**Prerequisites:** CS 523 or considerable experience in computer graphics; one of the courses CS 503 or CS 513 or equivalent.

**Outline:** Data structures and processors, display software, interactive graphic display system, pattern recognition, flowchart and block diagram processing, computer assisted instruction, chemical modelling and biochemical applications, computer aided design, mapping generation of moving pictures.

Students will be assigned tasks involving a digital plotter, microfilm plotter and an interactive graphic display system, and will undertake a major project in graphical information processing.

**Expected Work:** Tasks involving programming of a digital plotter and CRT plotters. Visits to computer graphics installations. Class reports on topics of interest. Work on a major project in interactive computer graphics display systems.

**References:**

July 1972
Computer Science 570    TOPICS IN COMPUTERS IN EDUCATION    (3 credits)

Main theme for the Spring of 1973:    EVALUATION OF CAI METHODS

Instructor    W. Fabens

Schedule    Spring semester    01    W    6:15-9:00

Objectives    To examine various approaches in the design and implementation of Computer Assisted Instruction (CAI) systems. The course contrasts the approaches to major issues in CAI (teaching strategies, centralization of equipment and methods of evaluation) as they are resolved in some existing CAI projects.

Prerequisites    CS 503 or CS513 or consent of the instructor.

Outline    Brief history of CAI; theories of how, what and why people learn; CAI authoring languages; automatic and semi-automatic teaching programs; the contrasts among Plato, Ticcit and other projects with respect to implementation, authorship, evaluation, subject areas and mode of instruction.

Expected Work    Weekly assignments, possible presentations and term projects.

References    Various research papers and articles.

July 1972
Computer Science 580  TOPICS IN COMPUTERS IN BIOMEDICINE (3 credits)

Instructor  
C. Kulikowski

Schedule  
Spring Semester, 01  M 6:15-9:00

Objectives  
This course will examine current problems in the application of computers to biomedical research. Emphasis will be placed on the description of biomedical systems in terms of models that can be realized on a digital computer.

The course is intended for computer science students interested in this area of research and for students in the biomedical sciences who are interested in advanced applications of computers in their work.

Prerequisites  
At least a basic course in probability and/or statistics (or equivalent experience), knowledge of a higher level programming language (Fortran, PL/I, etc.); undergraduate mathematics should include calculus and linear algebra. Students without the above requirements must see the instructor before registering for this course.

Outline  

Expected Work  
The student will be required to undertake a project in this course. It can range from a specific computer implementation to a broader investigation of one of the research areas. The student will present his results at one of the class sessions.

References:
Proceedings of the IEEE, Special Issue on Technology and Health Services, November 1969.

Readings will be assigned during the term, principally from the following periodicals: Computers and Biomedical Research, Data Acquisition and Processing in Biology and Medicine, Proc. of the IEEE, IEEE Transactions on Medical Electronics. IEEE Transactions on Systems Science and Cybernetics, JAMA, Radiology, Medical Documentation.

July 1972
Computer Science 583 - TOPICS IN DATA STRUCTURES (3 credits)

Instructor:  C. V. Srinivasan

Schedule:  Fall semester:  01 T 3:15-6:00 Hil-120

Objectives:  To discuss selected problems in the representation and manipulation of structured information; to explore new lines of research which may lead to work on a dissertation.

Prerequisites:  Atleast four graduate courses in Computer Science, including CS 503. Prospective students should consult with the instructor before registering for the course.

Outline:  Topics vary each year. The following topics were discussed in the Fall of 1970: Implementation of data structures in PL/I; Automatic data structure design procedures; File organizations; Sorting techniques; Question-answering systems; Virtual memory; Data structures for computer graphics; Approaches to Data Base design.

In the fall of 1971 we discussed the following: File Organization methods; cost and performance evaluation of various file organization schemes; hashing schemes and their application to large file organization systems; generalized data base systems; problem representations and heuristic search in problem solving systems; theoretical aspects of data organization.

In the fall of 1972 we plan to discuss Semantic information processing, coherent information systems with deductive capabilities.

Expected Work:  Homework assignments. One or more students will report to the class each week on a set of papers relevant to the current topic of interest. The rest of the group will be required to read at least one survey article on the week's topic. Each student shall also submit an essay on the topic reviewed by him. Instructor will introduce the topics to be discussed and principal problems in each area.

References:  Selected papers in the research literature. Also, the book "Semantic Information Processing" by Marvin Minsky. MIT Press.

July, 1972
Computer 590  TOPICS IN MEASUREMENT AND EVALUATION OF LARGE
COMPUTER SYSTEMS  3 credits

Instructor  D. N. Freeman

Schedule  Spring Semester: T 6:15-9:00

Objectives  Become familiar with the principal problems of measurement
and performance improvement for large, third-generation,
multi-programmed systems. Learn to propose appropriate
hypotheses for good/poor performance of hardware configura-
tions, as affected by operating systems and various types/
qualities of application programs. Create tools appropriate
to measure/tune these systems.

Prerequisites  198:504 and 198:519
Consent of the instructor

Outline  Review hardware and software elements of large, multi-
programmed systems (primarily general-purpose batch and
time-sharing services). Networks of large and small com-
puter will be studied in some detail.
Review hardware and software tools for measuring performance.
Costs, availability, current evaluation, and prognosis are
discussed at length.
Using the major Rutgers computer facility as a subject of
study, determine its current performance at macro and micro
levels, using Assembler-level subroutines to read out
hardware-activity and OS/360 activity. Tabulations, time-
series analysis, and causation inferences are given special
emphasis.

Expected Work  Selected readings of current papers (since 1968) in ACM
Communications, Datamation, JCC Proceedings, and the like.

Writing 5-10 Assembler-language programs to read out and
tabulate performance data.

Three-hour, closed-book, written final examination.

References  As cited above.

July, 1972
Computer Science 592 - TOPICS IN OPERATING SYSTEMS  (3 credits)

Instructor: William Easton

Schedule: Spring semester, 01 M 4:45-7:30

Objectives: Study in depth of selected areas of operating systems design. Intended for students interested in operating system design as a field of extended study or thesis work.

Prerequisites: CS 519 and consent of instructor.

Outline: Topics in operating systems of interest to students and instructor. Analysis and discussion of new design schemes and research problems.

Expected Work: Review papers and design projects.

References: Relevant papers are cited in CS 519. Selected papers in the research literature will be introduced in the course.

July 1972
Computer Science 593  TOPICS IN THE DESIGN AND IMPLEMENTATION OF
PROGRAMMING LANGUAGES I  (3 credits)

Instructor:  W. Fabens

Schedule:  Fall semester, T 3:15-6:00 P.M.  Records Hall 101

Objectives  This is the first of two courses, CS 593,594, intended to help
students gain insight and an understanding of the concepts
and techniques used in the design of programming languages,
taking into consideration the user's point of view and also
the problems of computer implementation. Each course can be
taken independently.

The course will be an examination of features of various
programming languages as they affect the user (rather than
the implementor).

Prerequisites  CS 515 or equivalent, as well as a knowledge of SNOBOL4;
or consent of the instructor. Knowledge of assembly
language is desirable, but not required.

Outline  The course will include a study of the specification of flow
of control, data objects and data operations, and syntax
that various languages allow. It will relate these methods
to the design of interactive languages especially.

Expected Work  Weekly assignments and possible term project.

References  Selected papers in the research literature.

July 1972
Computer Science 594  TOPICS IN THE DESIGN AND IMPLEMENTATION OF
PROGRAMMING LANGUAGES II  (3 credits)

Instructor:  I. Polonsky

Schedule  Spring semester:  01 Th 6:15-9:00

Objectives  Study of the design of programming languages through a detailed study of the implementation of a complex programming language.

Prerequisites  CS 515, knowledge of assembly language programming and SNOBOL4; CS 593 is desirable. Any of these may be waived with the consent of the instructor.

Outline  This course will consist of a detailed study of the implementation of a complex programming language such as SNOBOL4 or SPITBOL.

Expected Work  Weekly assignments and possible term project.

References  SPITBOL Manual

SPITBOL program Listing

July 1972
Computer Science 597 - TOPICS IN COMPUTER SYSTEM ORGANIZATION (3 credits)

Instructor            S. Y. Levy

Schedule              Fall Semester - 01 3:15-6:00 Hil-124

Objectives            To discuss selected problems and current research in computer organization. The course is recommended for advanced graduate students, and it may lead to work on a project or a dissertation.

Prerequisites          Knowledge of logical design and computing systems design; CS 507 and CS 508 or Electrical Engineering 558 and 561, or equivalent. Prospective students should consult with Instructor before registration.

Outline               Topics vary each year. In the Fall of 1971, the emphasis will be on the following topics: memory hierarchies, their characteristics, methods for their design and evaluation, and modes of utilizing them; parallel processing from a hardware point of view (e.g., parallel processors - Illiac IV, 360/2839, Advanced Avionic Computer, etc.), algorithms and special software for parallel processing; survey of new super-computer developments (e.g., CDC Star); and methods for monitoring and evaluating computer system performance.

Expected Work         Each student will report to the class a review of a selected set of papers, on a specific topic relevant to the subject of interest chosen for the semester. The topics, the associated papers for review, and the problems of interest in each topic will be introduced by the instructor. Students will be encouraged to propose and choose topics that interest them most, within the general area of discussion in the class. Each student should submit an essay on the topic reviewed by him/her.

References            Selected papers in the research literature will be introduced in the course.

July 1972
Computer Science 598  TOPICS IN ARTIFICIAL INTELLIGENCE (3 credits)

Instructor  S. Amarel

Schedule  Spring Semester - 01 T 4:45-7:30

Objectives  To study concepts, methods, and techniques in artificial intelligence, with emphasis on machine problem solving. Intended primarily for advanced graduate students in computer science. Exploration of current research in the field which may lead to work on a project or a dissertation.

Prerequisites  At least one course in theoretical aspects of computing (e.g., one of the courses CS 507, 528, 530, 539); one course in non-numerical applications or in optimization problems (CS 513 or equivalent, or CS 529, or CS 521, 522); knowledge of logic, algebra and combinatorials at least at the level of 198:205, 206 and Philosophy 332 (Intermediate Logic). In case of uncertainty about the requirements, students should consult with the instructor before registration.

Outline  The course will proceed by examining classes of problems that have received attention in artificial intelligence research; some of the problems will be studied in considerable detail to allow discussion of the main concepts involved, and others will be sketched and their relationship to other problems clarified. Emphasis will be placed on principles for the design of problem solvers, and connections with other areas in computer science. Subjects covered will include: Taxonomy of problems; structure of problem solving procedures; heuristic procedures; the problems of search and representation; production and reduction procedures; applications in theorem proving, syntactic analysis, reasoning about actions, scheduling; generality in problem solving; game playing and machine learning; question answering; formation problems (concept formation, hypothesis formation), and design problems.

Expected Work  Assignments: selected readings and critiques; analysis of problems and development of problem solving procedures (rationale for their design and specification up to overall flow chart level, sometimes up to working programs). Project: a comprehensive study of a research problem in the field.


An extensive bibliography will be distributed, and selected papers in the research literature will be introduced in the course.

July 1972
Computer Science 601, 602   SELECTED PROBLEMS IN COMPUTER SCIENCE

(variable credit each term)

General:

Description

This course is intended for advanced MS students and for intermediate level Ph.D. students. Students enrolling in the course should have a minimum of 12 graduate credits in computer science. The objective of the course is to study current problems of computer design and application, and to provide working experience in system implementations. The emphasis is on a "realistic" kind of problem solving by the student, in the sense of requiring a problem to be attacked as a whole, rather than in the form of a programming exercise or a library survey. Generally, some implementation will be a part of the problem, as well as literature searching and system design.

The topics will vary from term to term in accordance with the interests of faculty and students. The course may be organized around a single main theme, or it may develop around a collection of projects that are not necessarily related to each other.

The course provides a good opportunity for Ph.D. students to fulfill their requirements for pre-thesis project work.

Note: The credit for this course is variable. A student may elect to enroll for a pre-determined number of credits, which in effect determines the amount of work which will be expected; or he may choose to let the instructor assign the amount of credit at the end of the course based on the actual accomplished work.

(For more details see the following 2 pages.)
Computer Science 601, 602 (continued)

Main Theme in the Fall of 1972: REAL-TIME PROGRAMMING

Instructor E. J. Wilkens

Schedule Fall: 01 W 3:15-6:00 BE 013

Objectives This course will examine a specific class of systems which can be realized on a digital computer. The salient characteristics of this class of systems include one or more of the following:

- random or statistically distributed inputs from many sources,
- specified average or maximum response times,
- inputs from a single source which are distributed over periods of time which are long with respect to the response time,
- known, although typically complex, tasks to be performed in response to each input,
- responses to each input which are dependent on both the present input and past inputs.

Prerequisites CS 503 and CS 513, or equivalent.


Expected Work The student will be required to choose a specific type of system (process control, telephone, experiment control, etc.) which he will relate to the general models presented above. The work may range from analysis and comparisons of several systems to an implementation of a complete system or an optimization of one facet of an implementation.

References Selected papers in the research literature.

July 1972
Main Theme in the Spring: VISUAL INFORMATION PROCESSING

Instructor  G. Falk

Schedule  Spring Semester 01 T 6:15-9:00

Objectives  This course will concentrate on problems related to both the analysis and the synthesis of visual information. Emphasis, however, will be placed on analysis (CS 523 and CS 544 are concerned with synthesis in some detail). Ideas will be drawn from pattern recognition, computer graphics, and artificial intelligence. The course is intended primarily for computer science students interested in this area of research.

Special Prerequisites  CS 503 and CS 513 (or equivalents); in case of doubt about their qualifications for the course, students should consult with the instructor.

Outline  Topics to be considered will be among the following:

- Image preprocessing techniques.
- Internal representations (data structures) for 2 and 3 dimensional objects.
- Geometry relevant to scene analysis.
- Existing scene analysis systems - eyes for robots.
- Linguistic approaches in picture processing.
- Integrating analysis from different points of view.
- Analysis of scenes that change with time.

Expected Work  The student will be required to undertake a project. It can range from a specific computer implementation to an in-depth investigation of one of the research areas. Each student will present his results at one of the class sessions.


Other selected papers in the current research literature will be presented in class.

July 1972
Computer Science 611, 612  SEMINAR IN COMPUTER SCIENCE

(3 credits each term)

General: This course is intended for advanced graduate students (students who have a minimum of 18 graduate credits in computer science). The primary objective is to discuss areas of current research in computer science. The course provides a good opportunity for Ph.D. students to start work towards a dissertation. The main theme may change from year to year in accordance with the interests of faculty and students. Participation in this course (for at least one term) is required of doctoral candidates.

(For more details see following 2 pages)
Computer Science 611  SEMINAR IN COMPUTER SCIENCE  (3 credits)

Main theme in Fall 1972:  COMPUTATIONAL COMPLEXITY

Instructors  S. Levy, R. Orgass and M. Pauli

Schedule  Fall semester, 01 W 6:15-9:00 Hil-254

Objectives  To introduce students to current research on the analysis of algorithms.

Prerequisites  A knowledge of mathematics which includes elementary linear algebra and the basic algebraic structures (groups, rings, fields) will prove extremely useful. CS 513 is a recommended prerequisite.

Outline  The seminar will start with a brief survey of the major results and techniques in abstract complexity theory. However, most of the semester will be devoted to a study of concrete problems.

Sorting and merging (both internal and with tapes), optimal search methods, and operations on polynomials, matrices, trees and graphs will be discussed along with proofs of optimality where possible. Notions of relative difficulty and strategies for proving optimality will also be included if time permits.

References  No textbook will be used, but there will be current literature readings and instructor's notes.

July 1972
Computer Science 612  SEMINAR IN COMPUTER SCIENCE  (3 credits)

Main theme in Spring 1972:  CELLULAR AUTOMATA

Instructor  T. Ostrand

Schedule  Spring semester, 01 M 6:15-9:00

Objectives  To study the theory of cellular automata, and its application to pattern recognition, biological systems, and computer construction.

Prerequisites  CS 507 and one of: CS 508, CS 523, CS 528, CS 535, CS 539.

Outline

1. Theory
   Cellular system of von Newmann; iterative circuit computers of Holland; tessellation structure of Moore; tessellation automata of Yamada and Amoroso; bounded cellular spaces of Beyer and Smith.

2. Application
   Spatial computer of Unger; picture processing; biological studies of Baricelli; universal computer of Holland; cellular computer of Lee.

Expected Work  Reading and discussion of various papers in the literature. Each student will present one or more papers concerning a specific topic in cellular automata.


   Papers from recent literature.

July 1972
Members of the Computer Science Faculty and their research interests are listed below.

Saul Amarel; Professor, Chairman, and Director of Graduate Programs; B.S. (Technion, Israel); M.S., D. Eng. Sci. (Columbia); Artificial Intelligence, computer linguistics, theory of algorithms, computer applications and impact.

Stanley Baxendale; Associate Professor and Vice-Chairman; B.S. (Leeds), Computer graphics, automatic indexing, information systems.

Bertram C. Bruce; Assistant Professor; Ph.D. (University of Texas); Natural language processing, mathematical logic, artificial intelligence.

John T. Cox; Assistant Professor; B.S. (Purdue) M.E.E. (New York University); Time sharing, switching theory, automata theory, programming, computer aided instruction.

William B. Easton; Associate Professor; B.S. (Cornell), Ph.D. (Princeton), Operating Systems, Time sharing, language processors, mathematical logic.

William J. H. Fabens; Assistant Professor; B.A. (Cornell), M.S., Ph.D. (University of Wisconsin); Programming languages, interactive systems, instructional processes.

Gilbert Falk; Assistant Professor; B.S. M.S. (MIT), Ph.D. (Stanford); Artificial Intelligence, visual information processing, data structures, operating systems.

Fred Fender; Professor of Computer Science and Mathematics; B.S., M.S., Ph.D. (University of Pennsylvania). Numerical analysis, computer solution of differential equations.

David Freeman; Professor and Director of University Computing and Information Processing; Ph.D. (Cornell University), Operating systems, compiler construction, measurement of computer performance.

Michael Grigoriadis; Adjunct Associate Professor, Ph.D. (University of Wisconsin); Mathematical programming, operations research; modeling and simulation.

Kenneth R. Kaplan; Associate Professor; B.E.E., M.E.E., Ph.D. (Polytechnic Institute of Brooklyn). Simulation, modeling, machine organization, stochastic processes.

Casimir Kulikowski; Assistant Professor; B.S., M.S. (Yale); Ph.D. (University of Hawaii); Pattern recognition, decision processes, computers in biomedicine.
Saul Levy; Associate Professor; Ph.D. (Yeshiva University); Switching theory, theory of algorithms, computer architecture, theory of computational complexity.

Thomas H. Mott, Jr.; Professor and Dean of Graduate School of Library Service; A.B. (Rice), Ph.D. (Yale), Switching theory, programming theory, information systems.

Richard J. Orgass; Associate Professor; B.S. (City College, N.Y.); M.S., Ph.D. (Yale); Theory of Computation, applications of logic to computer science, theory of computational complexity.

Thomas J. Ostrand; Instructor, B.S. (Univ. of Pennsylvania) automata theory, language theory, software design.

Marvin C. Paull; Professor; B.S. (Clarkson), Theory of Programming languages, translators, switching theory.

Ivan P. Polonsky; Adjunct Professor; Ph.D. (New York University); Programming languages and processors, operating systems.

Irving N. Rabinowitz; Professor; B.S. (City College of N.Y.); M.A., Ph.D. (Princeton), Programming languages, language processors, software design.

Chitoor V. Srinivasan; Associate Professor; B.Sc., D.M.I.T. (Madras), M.S., D. Eng. Sci. (Columbia); Description Languages, data structures, computer organization, coding theory.

Robert Vichnevetsky; Professor; Ph.D. (Brussels University), Computer methods for ordinary and partial differential equations, approximation theory, optimization theory, modeling and simulation.

Edward J. Wilkens; Assistant Professor; Ph.D. (University of Pennsylvania), switching theory, automata theory, operating systems.

Ann Yasuhara; Assistant Professor; Ph.D. (University of Illinois), recursive function theory, automata theory, combinational systems and logic.

July 1972