DOCUMENT RESUME

ED 022 699 SE 005 091

By-Duren, William L., Jr.

RECOMMENDATIONS ON THE UNDERGRADUATE MATHEMATICS PROGRAM FOR WORK IN COMPUTING.

Committee on the Undergraduate Program in Mathematics, Berkeley, Calif.

Spons Agency-National Science Foundation, Washington, D.C.

Pub Date May 64

Note-33p.

EDRS Price MF-$0.25 HC-$1.40

Descriptors-*COLLEGE MATHEMATICS, *COMPUTER ORIENTED PROGRAMS, *COMPUTERS, COURSE DESCRIPTIONS, *CURRICULUM, CURRICULUM DEVELOPMENT, CURRICULUM PLANNING, MATHEMATICS, OBJECTIVES, UNDERGRADUATE STUDY

Identifiers-Committee on the Undergraduate Program in Mathematics, National Science Foundation

Presented is a program for the undergraduate mathematical preparation for work in computer science. Discussed are three types of courses. (1) mathematics courses of a general nature, which should be available for the prospective specialist in Computer Science, (2) technical courses in Computer Science, most of which will generally be taught outside the mathematics department, and (3) laboratory courses in Computer Science designed to acquaint the student with the scope and power of a high speed computer and with some of the techniques by which its potential can be realized. Separate programs for mathematics and for non-mathematics majors who plan to enter the field of computing are described. (RP)

C

MATHEMATICAL ASSOCIATION OF AMERICA

RECOMMENDATIONS ON THE UNDERGRADUATE MATHEMATICS PROGRAM FOR WORK IN COMPUTING

COMMITTEE ON THE UNDERGRADUATE
PROGRAM IN MATHEMATICS

MAY, 1964

The Committee on the Undergraduate Program in Mathematics of the Mathematical Association of America is charged with making recommendations for the overall improvement of college and university mathematics curricula at all levels and in all educational areas. The Committee, through its parent association, has received a grant from the National Science Foundation to support its work. To carry on the activities under this grant, the Committee has organized the Commission on the Undergraduate Program in Mathematics consisting of the members of the Committee, an Executive Director, and an Associate Director.

Membership of the Commission on the Undergraduate Program in Mathematics is as follows:

WILLIAM L. DUREN, JR.
University of Virginia
Chairman

RALPH P. BOAS
Department of Mathematics
Northwestern University

R. CREIGHTON BUCK
Department of Mathematics
University of Wisconsin

DAN E. CHRISTIE
Department of Mathematics
Bowdoin College

ROY DUBISCH
Department of Mathematics
University of Washington

SAMUEL EILENBERG
Department of Mathematics
Columbia University

DAVID GALE
Department of Mathematics
Brown University

SAMUEL GOLDBERG
Department of Mathematics
Oberlin College

LEON A. HENKIN
Department of Mathematics
University of California, Berkeley

EDWIN E. MOISE
Graduate School of Education
Harvard University

JOHN C. MOORE
Department of Mathematics
Princeton University

HENRY O. POLLAK
Mathematics and Mechanics Research Center
Bell Telephone Laboratories, Inc.

In its discussions with engineers, physicists, and other scientists, the CUPM Panel on Physical Sciences and Engineering has been impressed with the all-pervading influence of modern high speed automatic computing and with the need for well-trained people in this field. Although the field is evolving rapidly, it nevertheless seems advisable that some statement be made regarding the proper training of such people, particularly from the standpoint of mathematics. This report is such a statement.

The following is the membership of CUPM's Panel on Mathematics for the Physical Sciences and Engineering at the time this report was prepared:

R. J. WALKER
Cornell University
Chairman

R. CREIGHTON BUCK
Department of Mathematics
University of Wisconsin

E. U. CONDON
Joint Institute for Laboratory Astrophysics
University of Colorado

CHARLES R. DE PRIMA
Department of Mathematics
California Institute of Technology

CHARLES A. DESOER
Department of Electrical Engineering
University of California, Berkeley

T. P. PALMER
Department of Mathematics
Rose Polytechnic Institute

HENRY O. POLLAK
Mathematics and Mechanics Research Center
Bell Telephone Laboratories, Inc.

G. BALEY PRICE
Department of Mathematics
The University of Kansas

MURRAY H. PROTTER
Department of Mathematics
University of California, Berkeley

A. H. TAUB
Digital Computer Laboratory
University of California, Berkeley

G. MILTON WING
Sandia Corporation
Albuquerque

1. INTRODUCTION

A striking development of the past ten years has been the increasing percentage of college graduates whose careers are intimately connected with high speed computing. A recent CUPM study has indicated that considerably more than a quarter of the students graduating with a major in mathematics from a broad selection of colleges and universities state that they are now working in the computing field. Many more people also enter this field from undergraduate programs in engineering, commerce, science, etc. It is evident that we must be concerned with adequate preparation for these students, and especially with an appropriate mathematics program. In this report the Panel on Engineering and Physical Sciences of CUPM proposes a program to meet this goal.

During this past decade the relationship of mathematics to computing and to computing machines has changed significantly. When the modern digital computer was developed, it appeared as a unique and important engineering achievement. In the beginning, it was generally thought of as a device for solving complicated but essentially routine problems in numerical mathematics. However, the advent of the stored program and the accompanying growth in the understanding of its potential, as well as the striking improvements in components and circuitry, have together brought about a radical change in the theory of these devices and in the depth and variety of the problems that can be treated. Computers are now being used for such nonnumerical tasks as the simulation of various types of complex systems, the performance of complicated logical operations, and the design of new computers and computer systems.

Out of the solution of such problems as these has emerged a field of study called Computer Science, embracing such topics as numerical analysis, theory of programming, theory of automata, switching theory, etc. Computer Science is closely related to mathematics; indeed, numerical analysis is a branch of that subject, while many of the problems arising from computers are intimately associated with questions in combinatorial mathematics, abstract algebra, and symbolic logic. But an even more fundamental relationship also exists. The inherent structure of a computer forces the attendant disciplines to strive for the type of generality, abstraction, and close attention to logical detail that is characteristic of mathematical arguments. Research workers in Computer Science must have a knowledge of the spirit and techniques of mathematics. Although they do not need to be mathematicians, they must think like mathematicians.

For these reasons a university or college organization charged with the responsibility for research and training in Computer Science should be closely linked to mathematics within the institution. (It goes without saying that it should also be coordinated with groups charged with providing computation services to the academic community.) It is to the combined group of mathematicians and computer scientists that we direct our report.

2. OUTLINE OF THE REPORT

This report makes reference to three types of courses:

(a) Mathematics courses of a general nature, which should be available for the prospective specialist in Computer Science.

(b) Technical courses in Computer Science, most of which will generally be taught outside the mathematics department. Because this material is not widely familiar, sample outlines of these courses are included in Appendix A.

(c) A three semester hour introductory course in Computer Science. This is designed to acquaint the student with the scope and power of a high speed computer and with some of the techniques by which its potential can be realized. It should include some practice in programming within a specific language and some programs should be run if a machine is available. This course is fundamental in any study of Computer Science and is prerequisite to all technical courses in this field. In Section 3 we include additional information about this course and suggestions for specific content.

Section 4 describes a program for a mathematics major planning to enter the field of computing. This is basically a standard mathematics program, with a few appropriate modifications. We have not stated any requirement in geometry-topology, feeling that courses in probability and in numerical analysis deserve greater priority. In addition, we require at least two electives from a specified list of courses in Computer Science.

A mathematics program for students who enter the computing field from other areas is given in Section 5. For these students we recommend a minimum of 21 semester hours of mathematics. The remainder of his program will be determined by his major field of study but should include a suitable number of courses in Computer Science.

Finally, a word about the way in which these recommendations were constructed. As is our usual practice, the Panel, after some preliminary discussion, held a series of meetings with groups of persons in the computing field, including, in particular, representatives of the Association for Computing Machinery. (A series of papers on the computer science curriculum recently appeared in the Communications of the ACM, 7(1964)205-227. These have many points of contact with this report and are highly recommended as supplemental reading.) At these meetings we sought the advice of our consultants and tested our own ideas against their informed opinion. The final document, however, is to be regarded as our own judgment; it is particularly noteworthy that we agree unanimously with the consultants' major suggestion, the introductory course in Computer Science.

We are sincerely interested in comments on this report and invite interested individuals to communicate with us by writing to:

Committee on the Undergraduate Program in Mathematics
Post Office Box 1024
Berkeley, California 94701

3. INTRODUCTORY COURSE IN COMPUTER SCIENCE

Since modern automata such as computers are playing an increasingly important role in everyday life and in education, it is important that, early in his career, the undergraduate achieve some intellectual understanding of these devices and of methods of using them. It is therefore recommended that a three semester hour introductory course in Computer Science be offered. This course would serve a number of purposes:

(1) It would give students an appreciation of the powers and limitations of automata.

(2) It would develop an understanding of the interplay between the machine, its associated languages, and the algorithmic formulation of problems.

(3) It would train students to use a modern computer.

(4) It would enable the instructors in later courses to assign problems to be solved on the computer.

Although this course may often be taught by a member of the mathematics department, we do not intend that it be considered as part of the standard mathematics curriculum. Its role is to form a basis for the technical sequence in Computer Science and an introduction to this area for all persons interested in modern computers.

The following is an outline of such a course for an institution where students have ready access to a digital computer. The course is divided into three approximately equal parts:

(a) Description of a computer. In this part the student is given a basic understanding of the logical structure of a computer and the notion of algorithms. Binary arithmetic is discussed, functions of the parts of a computer are described, machine orders are illustrated, and enough of the order code is taught to enable the student to write simple but nontrivial codes.

(b) Description of a programming language. This part of the course teaches the student how to prepare a problem for machine solution using a programming language. An effort should be made to motivate the restrictions and conventions in the language as well as to relate the structure of the language to the actual machine code produced by it and the translator.

Some computing languages currently in use in such programs and for which teaching materials are available are listed in Appendix B.

(c) Problem solving. This part of the course emphasizes the fact that many problems can be formulated in a manner using iterative procedures. Flow diagrams may be exploited in this connection. In the case of mathematical problems, the properties of the approximations involved should also be discussed.

Machine problems, both of numerical and nonnumerical types, are assigned during the semester, with emphasis being given both to the efficient use of the computer and to the variety of possible methods of solving the problems. Sample problems are listed in Appendix C.

4. PROGRAM FOR MATHEMATICS MAJORS

The following program is recommended for a student majoring in mathematics and expecting to work or to study further in computing. Courses 1, 2, 4 and 5 constitute a recognized core for a mathematics major. The total requirements are in line with those at many institutions.

The list of electives can be regarded as a sample rather than as exhaustive. Most of these courses will probably be taught outside the mathematics department, and the availability of personnel will determine the offerings at any institution.

Comments on the listed courses and sample syllabi are given in Appendix A.

A. The introductory course in Computer Science is a require ment (3 semester hours).

B. Required mathematics courses.

1. Beginning analysis (12 semester hours)
2. Linear algebra (3 semester hours)
3. Probability and statistics (3 semester hours)
4. Algebraic structures (3 semester hours)
5. Advanced calculus (6 semester hours)
6. Numerical analysis (3 semester hours)

C. Electives (3 semester hours each). A minimum of six semester hours. With the possible exception of Course 8 (Logic), the Introduction to Computer Science is a prerequisite for each of the following:

7. Numerical analysis
8. Logic
9. Information processing
10. Machine organization
11. Theory of automata
12. Advanced programming
13. Combinatorics
14. Systems simulation

5. MATHEMATICS PROGRAM FOR OTHER MAJORS

Many students enter the computing field from areas other than mathematics; for example, from engineering (particularly electrical) and business management. The full program of such students will, of course, vary with the major fields of study, but they need a certain core of mathematics as a foundation for their specialization in computing. On the basis of discussions with computer specialists from a wide variety of fields, the following courses in mathematics are recommended:

1. Beginning analysis (12 semester hours)
2. Linear algebra (3 semester hours)
3. Probability and statistics (3 semester hours)
4. Algebraic structures (3 semester hours)

The introductory course in Computer Science is also a requirement.

Electives appropriate to the student's major field can be chosen from items 5 through 14 of the previous section.

Appendix A. DESCRIPTION OF COURSES

While we feel strongly about the spirit of the courses here outlined, the specific embodiments are to be considered primarily as samples. Courses close to many of these have been taught successfully at appropriate levels, and our time schedules are based on such experience.

Some of these courses are sufficiently common that approximations to complete texts already exist; others have appeared only in lecture form. In general, the references which accompany an outline are not intended as texts for the course but as indications of possible sources of material to be molded into the course.

The descriptions of Courses 1, 2, and 3 are essentially the same as those appearing in the CUPM Recommendations on the Undergraduate Mathematics Program for Engineers and Physicists.

1. Beginning analysis (12 semester hours).

As far as general content is concerned, this is a relatively standard course in calculus and differential equations. There can be many variations of such a course in matters of rigor, motivation, arrangement of topics, etc., and textbooks have been and are being written from several different points of view. Beyond the specific items mentioned below, we make no recommendations on these matters.

The course should contain the following topics:

a. An intuitive introduction of four to six weeks to the basic notions of differentiation and integration. This serves the dual purpose of filling in the student's intuition for the more sophisticated treatment to come and preparing for immediate applications to physics.

b. Theory and technique of differentiation and integration of functions of one real variable, with applications.

c. Infinite series, including Taylor series expansions.

d. A brief introduction to differentiation and integration of functions of two or more real variables.

e. Topics in differential equations, including the following: linear differential equations with constant coefficients and first order systems--linear algebra (including eigenvalue theory, see 2 below) should be used to treat both homogenous and nonhomogenous problems; first order linear and nonlinear equations, with Picard's method and an introduction to numerical techniques.

f. Some attempt should be made to fill the gap between the high school algebra of complex numbers and the use of complex exponentials in the solution of differential equations. In particular, some work on the calculus of complex valued functions of a real variable should be included in items b and c.

g. Students should become familiar with vectors in two and three dimensions and with the differentiation of vector valued functions of one variable. Such material can obviously be correlated with the course in linear algebra (see below).

h. Theory and simple techniques of numerical computation should be introduced where relevant.

2. Linear algebra (3 semester hours).

The purpose of this course is to develop the algebra and geometry of finite-dimensional linear vector spaces and their linear transformations, the algebra of matrices, and the theory of eigenvalues and eigenvectors.

Content

a. Finite-dimensional vector spaces (5 lectures).

Motivation and definition of such spaces over the real numbers. Dependence and independence, subspaces, dimension, and bases.

b. Linear transformations and matrices (7 lectures).

Linear transformations on a vector space and definition of a matrix. Change of basis and its relation to the algebra of matrices. Rank of a transformation and of a matrix.

c. Linear equations (11 lectures).

Systems of linear equations and related vector spaces. Determinants, minors, and the rank of a system. Adjoints and elementary transformations with application to the inverse of a matrix. Computational techniques for inverses.

d. Applications of matrices to problems of biology, economics, etc. (3 lectures).

e. Quadratic forms (7 lectures).

The notions of inner product and length and their relation to bases. Orthogonal projection, the Gram-Schmidt process and its geometric interpretation. Quadratic forms and the effect of change of basis. Reduction to diagonal form. Positive definite forms and their geometric interpretation.

f. Eigenvalues and eigenvectors (6 lectures).

Definition and characteristic polynomials. Relation to diagonalization of quadratic forms. Applications to numerical techniques, including the computation of eigenvalues and eigenvectors, particularly for symmetric matrices.

References

Finkbeiner, D. T. INTRODUCTION TO MATRICES AND LINEAR TRANSFORMATIONS. San Francisco: W. H. Freeman and Company, 1960.

Hohn, F. E. ELEMENTARY MATRIX ALGEBRA. New York: The Macmillan Company, 1957.

Kemeny, Snell, Thompson. INTRODUCTION TO FINITE MATHEMATICS. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1957.

Murdoch, D. C. LINEAR ALGEBRA FOR UNDERGRADUATES. New York: John Wiley and Sons, Inc., 1957.

Paige and Swift. ELEMENTS OF LINEAR ALGEBRA. Boston: Ginn and Company, 1961.

Thrall and Tornheim. VECTOR SPACES AND MATRICES. New York: John Wiley and Sons, Inc., 1957.

For numerical methods see:

Faddeev and Faddeeva. COMPUTATIONAL METHODS OF LINEAR ALGEBRA. San Francisco: W. H. Freeman and Company, 1963.

Householder, A. S. PRINCIPLES OF NUMERICAL ANALYSIS. New York: McGraw-Hill Book Company, Inc., 1953.

3. Probability and statistics (3 semester hours).

Item 3-a is first covered in the discrete case and then the process is repeated for the continuous case. This procedure has two advantages: first, the basic concepts and facts are introduced in the discrete case where the technical difficulties are least; second, the continuous case in many instances merely involves reinterpreting statements and proofs in terms of integrals rather than sums, thereby providing the student with a welcome review of the basic ideas.

Item 3-b emphasizes the nature of thinking in statistical inference.

Content

a. Probability theory (discrete case, 17 lectures; continuous case, 9 lectures).

Sample space, event, ramdom variable, function of a random variable. Probability, expectation, variance, moments, Chebychev's inequality. Joint distributions, transformations of joint densities, conditional probabilities, Bayes' theorem, independence. Bernoulli trials, combinatorics, binomial distribution. Poisson distribution. Normal law, introduction to the law of large numbers and the central limit theorem. Elementary Markov chains.

b. Introduction to statistical inference (13 lectures).

The formulation of statistical problems and the rationale behind the choice of statistical procedures. An introduction to estimation and sampling, with point and interval estimation. Elementary hypothesis testing, power of a test. Regression, a few examples of nonparametric methods.

References - Probability

Cramer, H. THE ELEMENTS OF PROBABILITY THEORY. New York: John Wiley and Sons, Inc., 1955.

Derman and Klein. PROBABILITY AND STATISTICAL INFERENCE FOR ENGINEERS. New York: Oxford University Press, 1959.

Feller, W. PROBABILITY THEORY AND ITS APPLICATIONS. New York: John Wiley and Sons, Inc., 1950.

Parzen, E. MODERN PROBABILITY THEORY AND ITS APPLICATIONS. New York: John Wiley and Sons, Inc., 1960.

Statistics

Bowker and Lieberman. ENGINEERING STATISTICS. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1959.

Brownlee, K. A. STATISTICAL THEORY AND METHODOLOGY IN SCIENCE AND ENGINEERING. New York: John Wiley and Sons, Inc., 1960.

Brunk, H. D. AN INTRODUCTION TO MATHEMATICAL STATISTICS. Boston: Ginn and Company, 1960.

Hoel, P. G. INTRODUCTION TO MATHEMATICAL STATISTICS. New York: John Wiley and Sons, Inc., 1962, 3rd edition.

Lindgren and McElrath. INTRODUCTION TO PROBABILITY AND STATISTICS. New York: The Macmillan Company, 1959.

4. Algebraic Structures (3 semester hours).

Any standard treatment of the theory of groups, rings and fields.

5. Advanced Calculus (6 semester hours).

Either an introduction to real variable theory or a more applied approach, depending on the clientele.

6. Numerical Analysis (3 semester hours).

This course and the next form a one-year unit and may be given as such with minor revision or permutation of topics. However, this first half could be taken as early as the latter part of the sophomore year since its prime prerequisites are beginning analysis and, perhaps, a concurrent course in linear algebra. For both courses 6 and 7 computer time should be available to the student so that appropriate elementary problems can be actually tried out. Thus, the introductory course in Computer Science is certainly a prerequisite.

Content

a. Solution of equations (6 lectures).

Functional iteration of (nonlinear) equations, including convergence theorems, error effects, analysis of special methods such as the methods of false position and of Newton; iteration for systems of equations, methods of Bernoulli, Sturm, Graeffe, etc. for finding roots of polynomial equations.

b. Polynomial approximations; interpolation and quadrature (18 lectures).

Weierstrass Theorem, Bernstein polynomials, Lagrange interpolation with error formulas, least squares, orthonormal systems relative to given weight functions; concept and analysis of best approximation relative to given criteria -- Chebychev polynomials, trigonometric approximations. Differencing, interpolation schemes, and formal difference calculus; quadrature formulas of the interpolation and Gaussian types with an analysis of error; numerical quadrature for improper integrals.

c. Initial value problems for ordinary differential equations (9 lectures).

Reduction to first order systems, Runge-Kutta, Adams, and other predictor corrector methods, elementary considerations of stability and round-off.

d. Matrix inversion and matrix eigenvalues (6 lectures).

A first treatment of such problems, to include Gaussian elimination and some iterative methods for inversion; Rayleigh quotients and power methods for obtaining the eigenvalues of symmetric matrices, including an analysis of convergence.

References

Faddeev and Faddeeva. COMPUTATIONAL METHODS OF LINEAR ALGEBRA. San Francisco: W. H. Freeman and Company, 1963.

Hamming, R. W. NUMERICAL METHODS FOR SCIENTISTS AND ENGINEERS. New York: McGraw-Hill Book Company, 1962.

Henrici, P. DISCRETE VARIABLE METHODS IN ORDINARY DIFFERENTIAL EQUATIONS. New York: John Wiley and Sons, Inc., 1962.

Householder, A. S. PRINCIPLES OF NUMERICAL ANALYSIS. New York: McGraw-Hill Book Company, 1953.

Stiefel, E. L. AN INTRODUCTION TO NUMERICAL MATHEMATICS. New York: Academic Press, 1963.

Todd, J. SURVEY OF NUMERICAL MATHEMATICS. New York: McGraw-Hill Book Company, 1962.

7. Numerical analysis (3 semester hours).

This course, with its principal theme of partial differential equations and elementary functional analysis, demands a reasonable amount of mathematical maturity, such as would be obtained from courses in advanced calculus and linear algebra and should, therefore, logically follow such courses.

Content

a. Matrix inversion and matrix eigenvalues (10 lectures).

   Review, tridiagonal matrices, including the methods of Givens and Householder. Iterative methods.

b. Ordinary differential equations, boundary value problems and eigenvalue problems (11 lectures).

   Finite difference methods, energy methods, min.-max.

c. Partial differential equations of second order (18 lectures).

   Topics selected from the following: classification, analytical solutions of well-posed problems for single equations; maximum principles for elliptic and parabolic equations, L_2-or energy-estimates as well as pointwise estimates of solutions; hyperbolic equations, domain of dependence; Fourier analysis and stability for constant coefficient equations, eigenvalues for elliptic equations, iterative methods for difference equations arising from partial differential equations.

References

Forsythe and Wasow. FINITE-DIFFERENCE METHODS FOR PARTIAL DIFFERENTIAL EQUATIONS. New York: John Wiley and Sons, 1960.

See also the references for Course 6 above.

8. Logic (3 semester hours).

An introduction to deductive logic and to the theory of computability. This course would also be of interest to students contemplating graduate work in mathematics or engineering, as well as to computer scientists.

Content

a. Propositional calculus with equality: test by truth functions, proof procedure by axioms (traditional), conjunctive and disjunctive normal form, proof procedure by inference rules which yields also a decision procedure, rules for equality (10 lectures).

b. Quantification theory: prenex form, miniscope form, proof procedure, completeness proof using the infinity lemma, decidability of $(x_1)\ldots(x_{n_1})(Ey_1)\ldots(Ey_n)M$, formulas for provability (10 lectures).

c. Applications: Peano axioms, primitive recursive definitions, formal proof of simple theorems from axiomatic set theory, group theory, arithmetic and geometry (9 lectures).

d. Turing machines and Post production systems: definitions of these through examples, Turing machines as computer programs, the halting problem and its application to the word problem for semigroups, relation between Turing machines and monogenic production systems, Post's simple form of Godel's theorem in terms of production systems (10 lectures).

References

Church, A. MATHEMATICAL LOGIC. Princeton: Princeton Press, 1956.

Kleene, S. C. INTRODUCTION TO METAMATHEMATICS. New York: D. Van Nostrand Company, Inc., 1952.

Mendelson, E. INTRODUCTION TO MATHEMATICAL LOGIC. Princeton: D. Van Nostrand Company, Inc. 1964.

Rosser, J. B. LOGIC FOR MATHEMATICIANS. New York: McGraw-Hill Book Company, 1953.

Stoll, R. R. AN INTRODUCTION TO LOGIC. San Francisco: W. H. Freeman and Company, 1963.

9. Information processing (3 semester hours).

A survey of the problems and techniques involved in handling large amounts of information, both of numerical and nonnumerical character. Parts a, b, and c relate mainly to business data and will occupy from half to two-thirds the available time, depending on the clientele. Items d and e give an introduction to the handling of information in the form of natural languages.

Content

a. Serial file processing: files and formats; characteristics of serial peripheral equipment for computers; file processing languages.

b. Searching and sorting: methods for file searching and sorting; use of random access files and multi-level memory complexes.

c. Applications. Data reduction: analysis of data arising from physical scientific experiments; processing of survey data (social science, telephone traffic, engineering records); analysis of written text. File maintenance and document preparation: business systems and procedures; file processing connected with payroll, inventory, and financial accounting. Real time applications: on-line inventory maintenance, industrial process control, weapons systems.

d. Automatic translation: structure of natural languages, translation schemes, problems of ambiguity and multiple correspondence.

e. Information retrieval: classification systems, analysis of content; organization of stored information; schemes for retrieving stored information.

References

Becker and Hayes. INFORMATION STORAGE AND RETRIEVAL. New York: John Wiley and Sons, Inc., 1963.

Gregory and Van Horn. AUTOMATIC DATA PROCESSING SYSTEMS. Belmont, California: Wadsworth Publishing Company, 1963, 2nd edition.

Ledley, R. PROGRAMMING AND UTILIZING DIGITAL COMPUTERS. New York: McGraw-Hill Book Company, 1962.

McCracken, Weiss and Lee. PROGRAMMING BUSINESS COMPUTERS. New York: John Wiley and Sons, Inc., 1959.

10. Machine organization (3 semester hours).

An introduction to the concepts involved in the logical design of computers and computer systems.

Content

a. Brief review of the computer and how it functions.

b. Number systems: radix conversion, binary coded decimal, signed numbers, floating-point arithmetic.

c. Boolean algebra: set theoretic approach, manipulation rules. Propositional logic approach, truth tables and their simplification. Boolean functions and the problem of canonical forms.

d. Realization of combinational circuits. Switches and relays; solid state devices; other technologies.

e. Sequential machines. State diagrams and equivalence. Digital computer elements, adders code converters, shift registers, accumulators.

f. Logical design of a simple digital computer. Systems design, study of sequential operations, arithmetic and control units, requirements of memory. Input and output. Balance of various parts of a computer.

References

Mealy, George. BELL SYSTEM TECHNICAL JOURNAL 34, 1955, p. 1045.

Muller, David E. 'The Place of Logical Design and Switching Theory in the Computer Curriculum.' COMMUNICATIONS OF THE ASSOCIATION FOR COMPUTING MACHINERY, 1, 1964, 222-225.

Phister, Montgomery, Jr. LOGICAL DESIGN OF DIGITAL COMPUTERS. New York: John Wiley and Sons, Inc., 1958.

Rosenbloom, Paul. THE ELEMENTS OF MATHEMATICAL LOGIC. Dover Press, 1950, Chapter I and II.

Von Neumann, J. COLLECTED WORKS, Volume 5, ed. by A. H. Taub. New York: Pergammon Press. Distributed in USA by Macmillan. Particularly those papers involving preliminary discussions of the logical design of computing machines.

11. Theory of automata (3 semester hours).

This course can be given as a junior or senior year course (or possibly as a first-year graduate course). It would presuppose a course in modern algebra; that is, it would presuppose that the students can recognize whether they have proved something or not and that their background includes a slight familiarity with such things as partitions, equivalence relations, proof by mathematical induction, isomorphisms, automorphisms, homomorphisms, the empty set, union, intersection, subsets, inclusion, etc.

Content

a. Definition of the notion of finite automata. Methods of going from one type of description to another. State tables, synchronous sequential circuits, etc.

b. What can a finite automaton do? Kleene's theorem on the representability of events.

c. Reduced forms for sequential machines.

d. Algebraic description of finite automata. Semigroups, partitions of semigroups, monomorphisms.

e. Other classes of machines: potentially infinite machines such as Turing machines, nondeterministic machines, probabilistic machines.

f. The theory of computability. Statement of fundamental theorems on recursive functions.

g. The existence of self-reproducing and self-repairing machines. The theory of Von Neumann and Myhill.

It is recognized that anyone teaching a course in automata will probably wish to delete some of these topics and add two or three of his own choosing, because the subject is so new that there is not complete agreement on which topics are the most basic. In particular, items e, f, and g may well be replaced by others.

References

Gill, A. INTRODUCTION TO THE THEORY OF FINITE STATE MACHINES. New York: McGraw-Hill Book Company, 1962.

Ginsburg, S. AN INTRODUCTION TO MATHEMATICAL MACHINE THEORY. Cambridge: Addison-Wesley Publishing Company, 1962.

McNaughton, R. THE THEORY OF AUTOMATA, in ADVANCES IN COMPUTERS, Volume 2. New York: Academic Press, 1961.

Shannon and McCarthy. AUTOMATA STUDIES. Princeton: Princeton University Press, 1956.

12. Advanced Programming (3 semester hours).

This course deals with various types of programming and serves to introduce students to the concepts involved in current work in this area. It is suggested that two weeks be devoted to the first part and that the remainder of the semester be divided evenly between the other two parts.

Content

a. Review. Assembly systems, method of storage allocation when using these, pseudo-orders, macros, modify and load techniques, monitor and executive systems.

b. Structure of languages. Study of a particular language such as ALGOL, its ambiguities, its method of dealing with recursions and procedures. List processing languages, compiler writing languages.

c. Theory of compilers. Nature of syntax directed compilers, compilers for dealing with problem oriented languages, compilers for dealing with compiler syntax languages. Discussion of the evolution of a translator from a simple language whose translator is given in machine language.

References

Iverson, K. A PROGRAMMING LANGUAGE. New York: John Wiley and Sons, Inc., 1962.

Sherman, P. PROGRAMMING AND CODING DIGITAL COMPUTERS. New York: John Wiley and Sons, Inc., 1963.

Schwartz, H. R. An Introduction to ALGOL. COMMUNICATIONS OF THE ACM, 5(1962)82-95.

13. Combinatorics (3 semester hours).

The object of this course is to introduce the student to combinatorics in a way which exemplifies the use of the material in other courses--chiefly groups and algebraic structures--culminating in an understanding of George Polya's fundamental theorem in enumerative combinatorial analysis. Riordan, John. An Introduction to Combinatorial Analysis. New York: John Wiley and Sons, Inc., 1958, and Ryser, Herbert J. Combinatorial Mathematics. New York: John Wiley and Sons, Inc., 1963.

Content

a. Permutations and combinations. A blend of Chapters 1 in the two texts; that is, set-theoretic terminology (Ryser) and generating functions (Riordan).

b. The principle of inclusion and exclusion. Chapter 3 of Riordan; Chapter 2 of Ryser.

Symbolic development of Riordan and permanents and their formulas from Ryser.

c. Generating functions. Chapter 2 of Riordan modified to give a better presentation of the algebra of sequences (as in the article 'Generating Functions' by Riordan to appear in Applied Combinatorial Mathematics, edited by E. F. Beckenbach). The section on derivatives of composite functions is an essential preliminary to cycle indicators to follow in the next chapter and to cycle indices in the theorem of Polya.

d. The cycles of permutations. Chapter 4 of Riordan. An essential preliminary to the cycle index.

e. The fundamental theorem of enumerative combinatorial analysis. Chapter 6, Section 8 et seq (at the teacher's discretion) supplemented by material from the following:

Solomon W. Golomb, A Mathematical Theory of Discrete Classification, INFORMATION THEORY, ed. by Colin Cherry, Butterworth and Company, London, 1961 (particularly valuable for introductory illustrations).

Frank Haray, The Number of Linear Directed, Rooted and Connected Graphs, TRANS. AMER. MATH. SOC., 78(1953)446-463.

John Riordan, The Combinatorial Significance of a Theorem of Polya, J. SOC. INDUST. APPL. MATH., 5(1957)225-237.

E. N. Gilbert and John Riordan, Symmetry Types of Periodic Sequences, ILLINOIS JOURNAL OF MATH., 5(1961)657-665.

N. G. de Bruijn, Generalization of Polya's Fundamental Theorem in Enumerative Combinatorial Analysis, NEDERL. AKAD. WESTENSCH. PROC. SER. A Vol. 62 = INDAG. MATH. 21(1959)59-69.

14. Systems Simulation (3 semester hours).

This course examines the subset of symbol manipulation applications of the computer that involves the numerical and logical representation of some existing or proposed system for the purpose of experimentation with the model and of comparison of proposed methods of operating the system. The primary purpose of the computer is thus not a calculating adjunct to experimentation but is the experimental medium itself.

A course in probability and statistics is a prerequisite.

Content

a. Programming languages designed for use in simulation, such as SIMSCRIPT, GPSS-II, DYNAMO.

b. Technical problems of simulation: synchronization of events, file maintenance, random number generation, random deviate sampling.

c. Statistical problems peculiar to simulation: sample size estimation, variance reducing techniques, problems of drawing inference from a continuous stochastic process.

d. Applications: queuing models; storage, traffic, and feed-back systems; design of facilities and operating disciplines.

References

Markowitz, Hausner and Karr. SIMSCRIPT: A SIMULATION PROGRAMMING LANGUAGE. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1962.

Tocher, K. D. THE ART OF SIMULATION. New York: D. Van Nostrand Company, Inc., 1963.

Appendix B. COMPUTER LANGUAGES FOR TEACHING

1. ALGOL. Numerous versions and modifications of ALGOL 58 and ALGOL 60 are in current use. Stanford University has been involved in the development of some of these and has used them in large-scale teaching operations. A good text is GUIDE TO ALGOL PROGRAMMING by D. D. McCracken. New York: John Wiley and Sons, Inc., 1962.

2. FORTRAN. This is in very wide use, and compilers are available for most computers. Of the two texts listed below, the first is a short but adequate introduction, the second a complete treatment.

A GUIDE TO FORTRAN PROGRAMMING, by D. D. McCracken. New York: John Wiley and Sons, Inc., 1961.

A FORTRAN PRIMER, by E. I. Organick. Reading, Massachusetts: Addison-Wesley Publishing Company, 1963.

3. MAD (Michigan Algorithm Decoder). A simplified version of ALGOL, developed primarily for use in teaching. Compilers are available for the IBM 704, 709, 7090, 7094, 7040, 7044. Manuals are obtainable from the University of Michigan Press. It is the language used in the very useful texts:

THE LANGUAGE OF COMPUTERS, by B. A. Galler. New York: McGraw-Hill Book Company, 1962.

INTRODUCTION TO DIGITAL COMPUTING, by B. W. Arden. Reading, Massachusetts: Addison-Wesley Publishing Company, 1963.

4. CORC (Cornell Compiler). A very highly simplified version of ALGOL, intended for rapid learning by students (and faculty) completely unfamiliar with computers, combined with an elaborate monitoring system. Compilers are available for the Burroughs 220 and the CDC 1604. Manuals can be obtained from the Cornell Campus Store.

Appendix C. SAMPLE PROBLEMS FOR THE INTRODUCTORY COURSE IN COMPUTER SCIENCE

Simple arithmetic

1. Tabulate x^n for x = 0(2)20, n = 1(1)5.

2. Make change from a $10 bill for a purchase of $a.bc.

3. Find all primes less than 1000

   (a) by division,
   (b) by the sieve of Eratosthenes.

Sorting

1. Sort and count a bridge hand.

2. Arrange N numbers in order.

Algebra

1. Linear interpolation in a stored table.

2. Solution of linear equations by elimination.

Analytic geometry

1. (a) Do two line segments in a plane (given by the cartesian coordinates of their end-points) intersect?

   (b) How many triangles are formed by N given line segments in a plane?

2. Find the vertices of the smallest convex polygon containing N given points in a plane.

Approximation

1. Numerical quadrature by the Trapezoidal Rule or Simpson's Rule.

2. Solution of algebraic equations by successive bisections of an interval or by another iterative process.

Symbol manipulation

1. Polynomial algebra. Represent a polynomial in one variable by the ordered set of its coefficients.

   (a) Linear combinations and multiplication.

   (b) The division transformation. The Euclidean algorithm. Obtain the highest common factor of two polynomials.

2. Find the shortest path from SLOW to FAST, changing one letter at a time, with intermediate words from a given list.

3. Artifical English. Working with a passage of text, start with a letter a_1 and let a_2 be the letter following the first occurrence of a_1, a_3 the letter following the next occurrence of a_2, etc. Count spaces as letters but ignore punctuation.

Miscellaneous

1. (a) Write all permutations of the digits 1 2 3 4 5 6.

   (b) Solve the traveling salesman problem for seven cities.

2. Given N positive numbers in nondecreasing order, list all distinct triads that can be the sides of a triangle.

3. Experimental probability. If a random number generator is available, find the expected length of a game of Russian Roulette or of a random walk in a line segment or a bounded plane region, the expected waiting time at a supermarket checkout, etc.