# Report to the National Science Foundation

# on Grant GJ-660*

Lloyd D. Fosdick
Editor
Department of Computer Science
University of Colorado
Boulder, Colorado

Report #CU-CS-032-73 December, 1973

*This work supported by National Science Foundation Grant GJ-660

Report to the National Science Foundation on Grant GJ-660

## I. INTRODUCTION

This report describes progress which has been made in the establishment of a computer science program at the University of Colorado at Boulder under NSF Grant GJ-660. In the three years since the Department was established, its faculty has grown to a size of thirteen members (fourteen full-time equivalent). Graduate degree programs leading to the MS Degree and Ph.D. Degree have been established with approximately eighty students enrolled in these programs in the last academic year (AY 1972-73). Active research programs in the areas of automata theory and formal languages, operating systems, validation of computer software, combinatorics, software modeling and numerical analysis are under way. This work is described in various publications and twenty-six technical reports of the Department.

## II. HISTORICAL NOTES

The Department of Computer Science was formally created in the spring of 1970 by the Board of Regents of the University of Colorado. Before that time, in fact, since 1962, there had been an interest in computer science at the University of Colorado which was focused in the activities of the Institute of Computing Science. There were approximately thirteen members of this Institute, all of whom were members of various faculties such as Electrical Engineering, Mathematics, and Philosophy. Shortly before the Department of Computer Science was created, a Master's program in computer science had been established under the auspices of the Institute of Computing Science and various courses at the Undergraduate and graduate level were offered. When the Department was created, the Institute was simultaneously dissolved and its educational program was incorporated in the Department. Five members of the institute became rostered in the Department of Computer Science. Some of these were half-time appointments and amounted to a total of four and a half FTE. At the time of its creation, the Department was located in the Graduate College. It was recognized that this was merely a temporary arrangment and instructions were given to the Department to select a permanent home in one of the existing colleges; presumably Engineering or Arts and Sciences. During the academic year of 1970-71, many discussions were held on this subject within the faculty and with the faculty of Mathematics. It became clear early in these discussions that there was strong sentiment favoring the creation of a separate school of Mathematical Sciences which would contain the Mathematics Department and the Department of Computer Science. However, after various conversations with the university administration, it became evident that such a proposal would not meet with their approval. The decision was then made by the Department to go into the College of Arts and Sciences, and the Department was formally established in the College of Arts and Sciences in 1971.

## III. INSTRUCTIONAL PROGRAM

The Department of Computer Science offers a number of courses at the undergraduate and graduate levels. A list of courses presently being offered by the Department is presented in Attachment I of this document. At the University of Colorado, courses carrying numbers in the 100's through the 400's are generally regarded as undergraduate courses and courses in the 500's and 600's are regarded as graduate level courses. However, graduate students can take selected 400-level courses, and conversely, undergraduates are permitted to take selected 500-level courses for credit.

## III.I Undergraduate Instructional Program

While the Department of Computer Science offers no undergraduate degree in Computer Science, there are a number of ways that an undergraduate student at the University of Colorado can pursue studies in this area. For those students who have an interest in the area of engineering, there is a program within the Electrical Engineering Department leading to a Bachelor's Degree in Electrical Engineering and Computer Science. Many of the undergraduate courses which we offer support this program. For those students who are more mathematically oriented, there is a program within the Department of Mathematics allowing a student to obtain a Bachelor's Degree in Mathematics with special emphasis in computing. Undergraduate courses which we offer in our Department support this program. Within the College of Arts and Sciences, students may take a distributed major or an individually structured major involving computer science. In the distributed major program, students take courses concentrated in two areas one of which may be computer science. Courses which we recommend for this program are listed in Attachment 2. The individually structured major is designed for those students who wish to concentrate in an area not commonly recognized as a regular degree program. Since this option is a new one for us, we have decided for the time being to handle students on an individual basis with respect to their programs of study. Since the establishment of the Department in 1970, the enrollment in our undergraduate courses has gradually increased, and in the last academic year the enrollment (by head count) in all courses we offered at the 400-level and below totaled 1403. It should be pointed out, that the University of Colorado has a total enrollment, which is restricted by law, to 20,000 students. This rule has been in effect for the last two years; consequently, any increments in our enrollment actually represent enrollment shifts with decrements taking place elsewhere. Under these circumstances, notwithstanding the present popularity of computer science, large changes in this enrollment figure are not anticipated.

As in most computer science departments, the problem of dealing with large numbers of students in the elementary programming courses is a subject of considerable attention and study. Our introductory programming course is, in fact, three distinct courses aimed at students in three different interest areas. Computer Science 201 is aimed at students in engineering and the physical sciences. This course is cross-listed with Electrical Engineering 201 and we typically offer two sections; one taught by a faculty member from Electrical Engineering and another taught by a faculty member from the Department of Computer Science. The lecture recitation section format is followed with two lectures offered each week and one recitation section meeting for one hour each week. The primary language of instruction is Fortran. Computer Science 202 is designed for students in the School of Business and for other students who have a particular interest in the business applications of computers. It, too, follows the lecture recitation section format with three lectures offered each week and a one-hour recitation section each week. In this course, the primary language is Cobol. Computer Science 203 is aimed at students in the Liberal Arts. It, too, has a lecture recitation section with three hours of lecture each week and one hour of recitation. The primary language is Fortran. It has been our experience that support for teaching assistants has been decreasing and we are forced to examine new ways to organize these courses. One thing we are experimenting with is heavier reliance on undergraduates who are hired on an hourly basis and who serve as consultants to students. Results of our first attempts in this direction have been somewhat mixed and not altogether satisfactory. However, we remain convinced that undergraduates can be used effectively for the support of instruction in these courses and we will continue working in this direction. While the turn-around time at the Computing Center has been exceptionally good, there is a large bottleneck in getting jobs into the machine because of a limited number of keypunches. Financial support for additional keypunches and physical space required to house them does not appear to be available at this time. Furthermore, there is some question in our own minds as to whether or not simply increasing the number of keypunches is really the right answer to this problem. We have been examining the possibility of using marksense cards or prepunch cards (Porta-punch cards) and this fall we plan to use Porta-punch cards with a small group of students drawn from Computer Science 201 and measure the effectiveness of this mode of program preparation. We have not made any serious attempt to introduce time sharing in courses at this level. However, we are planning to introduce time sharing on a limited basis to some of our upper division undergraduate courses (courses at the 300- and 400- levels). Since terminal facilities for doing this are just now being acquired, this will not begin until next year. As yet another approach to the problem of dealing with large numbers of students in the elementary programming course, we intend to explore the use of a tutorial system of instruction which might eliminate lectures almost entirely. This mode of instruction has been used successfully in other areas including Psychology and Mathematics, and we feel that it might be quite effective in our program. One thing that has slowed progress in implementing this is the availability of adequate physical facilities. Finally, it is worthwhile mentioning that we have instituted an undergraduate course which allows students to participate in some of our activities which require tutors and to receive undergraduate credit for tutoring and playing a role in developing course problems and special teaching aids. This program is too new for us to make a reasonable evaluation of its effectiveness at this time.

## III.II Graduate Instructional Program

The Department of Computer Science offers both the M.S. Degree in computer science and the Ph.D. Degree in computer science. The M.S. Degree program had already been initiated at the time the Department was created in 1970. Planning for the Ph.D. program began in the fall of 1970 and was officially approved by the end of 1971. The first students were admitted to the program in the spring semester of 1972. The Department decided to put a quota on the number of students admitted to the Ph.D. program and decided to tie the quota figure to the number of FTE faculty members. The quota is not regarded as totally rigid but the intention is to adhere to it fairly strongly and deviate from it only in very exceptional cases. Presently, our quota figure is based on one Ph.D. graduate student for every FTE in the Department, which establishes the present quota at thirteen. As implied by these remarks, the Department tries to make a careful distinction between those students enrolled in the M.S. Degree program and those students enrolled in the Ph.D. program. Those students who are accepted into the Ph.D. program are encouraged to work directly toward that degree and they do not normally receive an M.S. Degree. Students in the M.S. Degree program, on the other hand, are regarded as being interested only in the M.S. Degree and it is not presumed that they will enter the Ph.D. program upon completion of the M.S. Degree work.

III.II.I The description of our M.S. Degree program is contained in Attachment 3. Students pursuing the M.S. Degree may do so under the thesis or nonthesis option. The nonthesis option, which was introduced only a year ago, is increasing strongly in popularity. Under this option, students select a reading list of papers in one of several different areas. They study the papers on this list over the course of one semester and take an oral examination on the material at the end of that semester. One novel feature of this is that about the middle of the semester the students are given a "warm-up" oral in which questions are presented to them on the material which they have studied so far and the students are advised as to their performance. However, this "warm-up" examination is not graded and is used only to acquaint the student with the examining procedure. Students in the nonthesis, as well as students in the thesis option, are asked to take a comprehensive examination covering elementary aspects of computer science. A sample set of reading lists is presented in Attachment 4. A list of M.S. thesis topics is presented in Attachment 5. Finally, a list of students who have been awarded the M.S. Degree in computer science is shown in Attachment 6. Also shown here are the present occupations of those students.

The department has recently completed plans for a business oriented computing curriculum within our M.S. degree program. Students entering this curriculum are assumed to have a Bachelor's degree in Business and to have taken some computing courses (10 hours) and upper division mathematics (9 hours). In the M.S. degree program they take a set of courses which the Department of Computer Science and the School of Business has agreed would be appropriate. Further details appear in attachment 7.

### III.II.II Ph.D. Program

As noted above, the Ph.D. program has limited enrollment. At the present time, in the academic year 1972-73, a total of 7 students were enrolled in the Ph.D. program. Students in the Ph.D. program are required to take a preliminary examination at an early date. This examination is given either during the first semester of their enrollment in the program (the fall of the academic year), or in those cases where it is thought that the student is ill-prepared, may be deferred until the start of the second year. A novel aspect of the preliminary examination is that, rather than being a "one shot" affair, it is a continuing exam which takes place over the course of one semester. The typical pattern of this exam is that a meeting is held once a week approximately two and one-half to three hours in length in which all students presently taking the preliminary exam are present and a minimum of two, usually three, faculty members are present. The students are presented with a wide range of problems and they are usually given a week to two weeks to prepare solutions to these problems. These problems may be open problems; in general they are very difficult. A second part of this preliminary examination is written and covers basic aspects of computer science. The written examination is primarily a test of knowledge of basic facts while the oral is primarily a test of problem solving ability.

Research programs occupying students in the Ph.D. program include software validation, portable software design, and automata theory.

## Research Programs

The research activities of the department are focused in the areas of automata theory and formal languages, operating systems, validation of computer software, combinatorics, software modeling and numerical analysis. An overview of research activities of individual faculty members can be obtained from the brief statements below.

Daniel Bailey. Professor Bailey has been concerned with problems of interactive computing, especially as it relates to a psychological testing environment. He is also concerned with problems in the area of statistical analysis of psychological data.

Andrzej Ehrenfeucht. Professor Ehrenfeucht has been working on the application of combinatorial methods to the abstract theory of learning. He has also been working on abstract languages, specifically OL-Languages (Lindenmayer - languages).

Clarence Ellis. Professor Ellis has been working on models of computer systems, specifically: multi-multi-processor computer systems; associative control systems; and a probabilistic automaton model and simulation of deadlock in computer systems.

Richard E. Fairley. Professor Fairley has been working on problems of semantic modeling of programs, including, specifically, semantic models of parameter passing. He is also working on an interactive program testing system, and problems related to the analysis of execution paths in computer programs.

Lloyd D. Fosdick. Professor Fosdick has been working on problems related to the validation of mathematical software. These include the analysis of control paths in programs, automatic generation of test data, and searching algorithms on graphs.

Robert C. Gammill. Professor Gammill has been working on a portable compiler generator system, contouring algorithms, and forcing strategies and openings of a simple game (4x4x4 tic-tac-toe).

John M. Gary. Professor Gary has been studying problems related to the solution of partial differential equations. These problems include: computational efficiency of difference schemes of higher order accuracy; extension of the Adams-Moulton algorithm or similar algorithms for ordinary differential equations to obtain a difference scheme which is fourth order in space and time; construction of an iterative method yielding a difference scheme which is fourth order in space and second order in time.

John S. Maybee. Professor Maybee has been working on problems in matrix theory and combinatorics arising from problems in mathematical economics. He has also been working on stability, and vibration problems.

Gary J. Nutt. Professor Nutt is concerned with problems in the area of measurement and evaluation of computer systems. In this connection he has been working on evaluation nets for the representation of parallel systems and on applying automata theory to the deadlock problem.

Leon J. Osterweil. Professor Osterweil has been concerned with problems in combinatorics, including graph enumeration and searching, problems in the validation of mathematical software, stochastic modeling of multiprocessor access to interleaved memories, and certain aspects of parallel processing computer systems. He has also been involved with urban modeling efforts.

H. Paul Zeiger. Professor Zeiger has been working on some problems related to regular expression representations, specifically, using regular expressions to represent programs and the construction of an "efficient" regular expression for a given regular event. He has also been working on the problem of obtaining an internal description of a linear dynamical system given approximate external data.

The research work of the Department is also represented in the list of recent publications and the list of technical reports (attachments 8 and 9). Outside support of our research activities is listed in attachment 10.

## Interdepartmental Activities

The faculty of the department is enriched by the presence of a number of other faculty members (courtesy faculty members) who do not occupy budgeted positions in the department but who enjoy the rights and privileges of a faculty member. A list of courtesy faculty members is attached (attachment 11). As can be seen from this list there is a representation from a variety of areas including electrical engineering, mathematics, management science, data processing, and physics.

Laboratory facilities in other departments are made available to students in computer science. These include computer laboratories in electrical engineering and psychology.

As noted above there are instructional programs in other departments, Electrical Engineering, Mathematics, and in the School of Business which are closely coupled to our own programs.

Interaction with Government Laboratories, Industry and Other Outside Agencies.

In the Boulder-Denver area there are a number of agencies engaged in computer oriented work. Many of our graduate students are employees of these agencies. A list of these employee-students is attached (attachment 12).

The department has just initiated an internship program with the Hewlett-Packard Corporation in Loveland, Colorado. In this program students work half-time at the plant and go to school half-time. They receive six hours graduate credit for a project they engage in connected with their work at the plant. We are hopeful that we will be able to extend this program to other companies and government laboratories. A copy of the memorandum of agreement between the Hewlett-Packard Corporation and the University of Colorado for this program is attached (attachment 13).

# COURSES OFFERED BY
# THE DEPARTMENT OF COMPUTER SCIENCE
# UNIVERSITY OF COLORADO

CS 162-1. Digital Computer Programming.

An introduction to digital computer programming with emphasis on mathematical applications.

Prer., High school algebra or equivalent.

CS 201-3. Introduction to Computer Science for Engineers and Scientists. (EE 201)

An elementary course in computer science covering computer programming methods, FORTRAN programming, numerical and non-numerical applications. Applications discussed will be chosen to reflect the interests of scientists and engineers.

Prer., Second year high school algebra.

CS 202-4. Introduction to Computer Science for Business Majors.

An elementary course in computer science covering computer programing methods, and COBOL programming. Applications discussed will be chosen to reflect the interests of Business majors.

Prer., Second year high school algebra.

CS 203-4. Introduction to Computer Science for Liberal Arts Majors.

An elementary course in computer science covering computer programming methods, FORTRAN programming, numerical and non-numerical applications. Applications discussed will be chosen to reflect the interests of Liberal Arts majors.

Prer., Second year high school algebra.

CS 290-1-6. Special Topics in Computer Science (Lower Division).

Selected topics at the elementary level for students who have had little or no previous computing experience.

CS 311-3. Computer Applications in the Mathematical Sciences.

An advanced FORTRAN programming course for scientists and engineers. Emphasis on the use of computers as computational tools in engineering and in science. Programming of typical mathematical applications.

Prer., CS 201, 202, or 203, or equivalent.

CS 312-3. Business Data Processing Methods.

This course is designed to introduce the student to some of the prevalent problems in business data processing. The student will study FORTRAN and its use in the implementation of some methods for handling these problems. Other languages appropriate to business data processing, such as COBOL, RPG, and BAL may also be discussed.

CS 313-3. Computers in the Social and Behavioral Sciences and the Humanities. (Psy 313)

This course presents computer applications in the social and behavioral sciences and the humanities. The emphasis of the course is on programming and use of the computer rather than on the substantive problems and issues.

Prer., CS 201, 202, or 203.

CS 400-3. Assembly Language and System Software.

Computer structure, machine language, instruction execution, addressing techniques, and digital representation of data. Data structures, subroutine linkages, reentrant code, and macros. Assemblers, linkers, and loaders. Job control and file systems. I/O systems, and alternate architecture.

Prer., FORTRAN.

CS 401-3. Introduction to Programming Languages and Processors. (EE 401)

A study of programming languages and digital processors. Conceptual aspects of programming languages, translators, data structures, hardware organization and systems architecture. Relationship of language features to processor features.

Prer., CS 201, 202, or 203, or equivalent.

CS 402-3. Hardware Structure and Software Design.

A course designed to present an overview of computer hardware concepts to software oriented people. The course deals with individual hardware devices and large scale hardware systems.

Prer., CS 400, or CS 453.

CS 453-3. Assembly Language Programming. (EE 453)

A laboratory course in programming at the machine code level. Lectures deal with the organization of the machine, its effect on the order code, techniques for programming in Assembly Language. Primary emphasis on preparing and running programs.

Prer., CS 201, 202, and 203.

CS 457-3. Logic Circuits. (EE 457)

A study of Boolean algebra, and its application to the syntheses of logical circuits from logical elements such as and-gates, or-gates, not-gates, nand-gates, nor-gates, delay elements, and memory elements.

CS 459-3. Computer Organization. (EE 459)

This course is concerned with computer arithmetic units, memory systems, control systems, and input-output systems. The emphasis is completely on logic structure rather than electronic circuitry.

Prer., CS 457 or equivalent.

CS 465-3. Intermediate Numerical Analysis I. (M 465)

Solution of algebraic and transcendental equations. Solution of linear and nonlinear systems or equations. Interpolation, integration, solution of ordinary differential equations least squares, sources of error, and error analysis. Computer implementation of numerical methods.

Prer., CS 162 and AM 232, or M 313.

CS 466-3. Intermediate Numerical Analysis II. (M 466)

Continuation of CS 465; further development of same topics and introduction of new topics, such as matrix eigenvalue problems and summation of infinite series.

Prer., CS 465.

CS 490-1-6. Special Problems in Computer Science (Upper Division).

This course provides opportunities for independent study, work on a small research problem, or tutoring of lower division computer science students.

Prer., CS 201, 202, 203 or equivalent.

CS 531-3. Formal Languages.

An introduction to linguistics from the point of view of the computer scientist and mathematician. Formal and natural languages are examined in terms of their characterization and analysis via mathematical and computational means. Computer methods relevant to language processing. Programming assignments in syntactic and semantic analysis.

Prer., CS 546 and CS 555.

CS 540-3. Computer Decision Modeling. (Mg Sc 625)

Application of the methods of computing science to problems in management decision making. Emphasis is placed on simulation as a method for studying the behavior of dynamic systems and the use of optimization models for their control.

Prer., course in statistics.

CS 545-3. Data Structures.

This course covers techniques for representation and manipulation of structured information in a digital computer. Topics include linear lists; sequential and linked storage allocation; multi-linked and multi-dimensional lists; circular lists. Trees, traversal algorithms; representation and mathematical properties of trees. Dynamic storage allocation and garbage collection.

Prer., knowledge of FORTRAN (CS 201, 202 or 203).

CS 546-3. Theory of Automata.

The idea of computability will be discussed. Computational models ranging in complexity from a Turing machine up to a caricature of a modern digital computer. The formal theory of languages and its relation to these computational models will be explored.

Prer., consent of instructor.

CS 553-3. Fundamental Concepts in Programming Languages. (EE 553)

A study of the concepts which underlie the design of a programming language: basic operators, operand modes and formation rules. Examples are drawn from contemporary languages such as FORTRAN, COBOL, ALGOL, LISP, SNOBAL, and PASCAL. The relationship among languages, hardware and applications is studied in the light of the underlying concepts.

Prer., CS 201, and CS 453.

CS 554-3. Numerical Applications of Digital Computers. (EE 554)

A survey of the types of digital computers useful for the execution of algorithms, for the solution of typical numerical problems encountered in science and engineering.

Prer., CS 453 (EE 453) and CS 311 or CS 465 or equivalent.

CS 555-3. Non-Numeric Techniques for Digital Computers (EE 555)

A study of the methods used in implementing processors for non-numeric problems; dynamic storage allocation, list processing, recursive programming and string manipulation. Several special purpose languages and their implementations will be studied in detail.

Prer., CS 453 and CS 401 or equivalent.

CS 556-3. The Translation of Programming Languages. (EE 556)

A study of problems of translating text generated by humans into programs understood by machines: Assemblers, fundamental concepts in programming language design, translation of ALGOL, code optimization in FORTRAN, syntax-directed techniques.

Prer., CS 453 and CS 401 or equivalent.

CS 557-3. Operating Systems. (EE 557)

A study of the means by which programming systems are integrated into the overall operations of a computing facility. Program segmentation and loading filing systems, resource allocation.

Prer., CS 401 or equivalent.

CS 558-3. Artificial Intelligence. (EE 558)

The design of machines and systems that have been created to perform tasks that are considered to require intelligence.

Prer., CS 162 or 201 and CS 401.

CS 559-3. On-Line Computing Systems.

Principles of on-line hardware and software. Survey of applications, on-line systems architecture, interrupt and I/O systems, real time operating systems, and data structures. Data communications hardware and software. Terminals and other peripheral devices. Queuing models. On-Line system analysis and design.

Prer., FORTRAN and Assembly Languages.

CS 560-3 . Numerical Analysis I. (Math 565)

Solution of linear systems, least squares approximations, solution of non-linear algebraic equations interpolation, quadrature, numerical methods in ordinary differential equations.

Prer., AM 442, Math 313, CS 162 or equivalent.

CS 561-3 Numerical Analysis II. (Math 566)

Solution of ordinary and partial differential equations, matrix eigenvalue eigenvector problems.

Prer., Math 565.

CS 562-3. Numerical Solution of Ordinary Differential Equations.

Convergence and stability of single step and multi-step methods. Stiff equations. Matrix and shooting methods for two-point boundary value problems.

Prer., CS 465, Math 313, Math 443, Math 431, or Math 435.

CS 563-3. Numerical Solution of Partial Differential Equations.

Stability and convergence for difference schemes. Leapfrog, Lax-Wendroff, implicit, ADI methods. SOR, ADI, direct methods for elliptic equations. Galerkin methods.

Prer., CS 465, Math 313, Math 431, or Math 435, Math 443.

CS 564-3. Numerical Linear Algebra.

Direct and iterative solution of linear systems. Eigenvalue and eigenvector calculation. Error analysis. Reduction by orthogonal transformation.

Prer., CS 465, Math 313.

CS 565-3. Numerical Methods for Optimization.

Linear programming. Unconstrained minimization, one dimensional search, gradient methods. Nonlinear and quadratic programming.

Prer., CS 465, and Math 313.

CS 566-3. Numerical Data Analysis.

Least squares fitting. Singular value decomposition. Fourier analysis of data. Surface fitting. Applications.

Prer., CS 465, Math 313, and Math 481 or 482.

CS 580-3. 581-3, 582-3. 583-3, Topics in Computer Science.

Selected topics in Computer Science covering theory of computers and computing, mathematics of computing, and application of computers to primary academic fields.

CS 590-1-6. Special Problems in Computer Science (Graduate)

This course provides opportunities for independent study and for work on research problems. Course content to be arranged with instructor.

CS 611-3. Topics in Computer Graphics.

Computer graphics hardware: printers, incremental plotters, microfilm, storage CRTs, and refreshed CRTs. Computer graphics software: special and general purpose subroutine packages, graphics languages, and data structures. Special problems: perspective viewing, hidden lines, windowing, & man-machine engineering.

Prer., Advanced programming ability.

CS 612-3. Topics in Operating Systems.

Selected topics in operating systems for digital computers. Topics to be selected by the instructor. Possible topics are: system design, measurement and evaluation, simulation, mathematical modeling, and parallelism.

Prer., CS 557.

CS 613-3. Topics in Programming Languages.

Selected topics in programming languages for digital computers. Topics to be selected by the instructors. Possible topics are: syntax, semantics, meta-compilers, compiler design and translator writing systems.

Prer., CS 401.

CS 614-3. Topics in Computer Systems.

Selected topics in computing systems. Topics to be selected by the instructor. Possible topics are: on-line systems, multi-processing, microprogramming, architecture, data communications, and computing networks.

Prer., CS 402.

CS 615-3. Topics in Formal Systems.

Selected topics in formal systems. Topics to be selected by the instructor. Possible topics are: Formal languages, abstract machines, combinatorics, finite mathematics and algorithm analysis.

CS 616-3. Topics in Data Processing.

Selected topics in data processing. Topics to be selected by the instructor. Possible topics are: Computing Center Management, Information Retrieval, Data Base Security, Commercial Systems, System Evaluation and Selection, Economics of Large Scale Systems.

CS 665-3 Advanced Numerical Analysis I.

Prer., Math 566.

CS 666-3. Advanced Numerical Analysis II.

Prer., Math 665.

CS 700-4-6 Master's Thesis

Students pursuing the Plan I thesis option will enroll for 4 to 6 credit hours of CS 700 while working on the thesis.

CS 701-3 Master's Reading Option.

An alternative to the Master's Thesis. Students will read selected papers and pass an examination.

CS 710-3. Doctoral Preliminary Seminar.

A seminar course required of all doctoral students. Students will demonstrate their research ability by investigating and reporting on current and open problems in Computer Science.

NOTE: 400 and 500 level courses require upper division standing. Students not meeting this requirement or students failing to meet the indicated prerequisites may have them waived with the consent of the instructor.

Courses of special interest in computer applications outside of the Department of Computer Science are: Psych 403, 503, Soc 523, and Soc 529.

Attachment 2 8/13/73

# Undergraduate Computer Science at the University of Colorado

The University of Colorado does not offer an undergraduate degree in Computer Science. However, the Computer Science Department participates in the Distributed Major Program of the College of Arts and Sciences. Undergraduate students in Arts and Sciences can satisfy the distributed major requirements by completing 30 credit hours in Computer Science and 30 credit hours in one or two secondary fields of study.

The suggested Computer Science courses are:

| | |
|---|---|
| CS 201 | Introduction to Computer Science |
| CS 311 | Computer Applications in the Mathematical Sciences |
| CS 400 | Assembly Language and System Software |
| CS 401 | Introduction to Programming Languages |
| CS 402 | Computer Architecture |
| CS 457 | Logic Circuits |
| CS 465 | Intermediate Numerical Analysis I |
| CS 466 | Intermediate Numerical Analysis II |
| CS 545 | Data Structures |
| CS 546 | Automata Theory |

One of the secondary areas should be Mathematics. The following courses are suggested:

| | |
|---|---|
| Math 130 | Analytic Geometry and Calculus I |
| Math 230 | Analytic Geometry and Calculus II |
| Math 240 | Analytic Geometry and Calculus III |
| Math 313 | Introduction to Linear Algebra |
| Math 481 | Introduction to Probability Theory |
| Math 352 | Computable Functions |

Each program of study will be individually arranged by the student and the Computer Science advisor.

Each student in the program must have an approved program of study on file in the Computer Science Department.

rcm

# MASTER OF SCIENCE
# COMPUTER SCIENCE DEGREE PROGRAM
# UNIVERSITY OF COLORADO

## Introduction

### Computer Science at the University of Colorado

The Computer Science Program at the University of Colorado includes the following areas of study: Automata Theory, Computing Systems Design, and Numerical Analysis. A graduate student should consider a major in computer science if he is primarily interested in the general aspects of computational processes, both theoretical and practical, e.g., theory of algorithms, methods by which algorithms are implemented on a computer, and information structures. If he is primarily interested in the results of a computer process and its relation to a particular area of application, then he should major in another field and consider a minor in computer science. Computer science courses which are application oriented will be taught as service courses to students of computer science and other fields.

Computer facilities at the University are extensive, including, on the Boulder campus, two CDC 6400's, a Librascope L-3055, an IBM 360/40, an XDS Sigma 3, a Beckman/XDS hybrid, and several Novas.

### Admission to Graduate Work in Computer Science

An applicant will be favorably considered for graduate study in computer science if he holds at least a bachelor's degree or its equivalent from an institution comparable to the University of Colorado. He should have some programming experience and sufficient mathematical maturity to understand pure mathematics courses at the level of a senior course in advanced calculus. The programming experience should be the equivalent of at least nine hours in under graduate computer courses (e.g., CS 201, CS 311, and CS 400 at the University of Colorado). In mathematics, he should have courses in integral and differential calculus, and three hours in each of the following subjects: linear algebra, probability theory, and advanced calculus. These mathematics requirements are interpreted as follows: for linear algebra, any upper division algebra course is acceptable;

for probability theory, any course in the areas of probability or statistics is acceptable; for advanced calculus, any course in advanced calculus, differential equations, logic, or any other course requiring evidence of mathematical maturity is acceptable. All courses referred to here should be at the junior or senior level and offered by a Department of Mathematics.

Due to the broad range of undergraduate majors from which computer science draws applicants for graduate level work some latitude in prerequisities is allowed. Six semester hours in computing (e.g., CS 201 and CS 311) together with integral and differential calculus are firm prerequisites. A student lacking the three remaining hours in computing (e.g., CS 400) must take them while on provisional degree status. A student lacking up to six of the nine semester hours in mathematics beyond integral and differential calculus must take them while on provisional degree status. The above noted work taken as a provisional degree student will count in credits for the M.S. degree. However, any additional hours which may be lacking are regarded as a deficiency which must be removed by taking courses for no credit.

Applicants should have a grade point average of at least 3.0 (on a scale of 4.0). Applicants having the listed qualifications will, if accepted, be classified as regular degree students. Applicants with an average below 3.0 and above 2.75 and/or lacking certain of the prerequisites, as indicated, will be considered on an individual basis. Students accepted in this category will be classified as provisional degree students. Though it is not a firm requirement, all applicants should submit Graduate Record Examination scores. This is especially true for applicants with a grade point average near or below 3.0, as G.R.E. scores weigh heavily in deciding borderline cases.

## General Description of Areas of Study

Each area of study, with lists of regularly scheduled courses which are relevant to these areas, is described below. In addition to these courses, there are additional opportunities for study in the CS 580 series (Topics in Computer Science), and in CS 590 (Individual Study in Computer Science).

a) Numerical Analysis and Computation

Students choosing to specialize in this area are expected to gain a cursory knowledge of computer organization and the inner workings of a computer and to develop a good understanding of linear algebra and elementary mathematical analysis. They would become proficient in the analysis and use of the basic methods of computational mathematics. This includes understanding the practical aspects of utilizing a computer with (necessarily) limited precision as well as the theoretical nature of standard numerical procedures. The principal areas of study should include computational linear algebra, interpolation and approximation, numerical solution of ordinary and partial differential equations, integration and differentiation, solution of algebraic and transcendental equations, and roundoff and truncation error analysis.

Courses available include:

CS 465-466 Intermediate Numerical Analysis I, II
CS 554 Numerical Applications of Digital Computers
CS 560-561 Numerical Analysis I, II
CS 562 Numerical Solution of Ordinary Differential Equations
CS 563 Numerical Solution of Partial Differential Equations
CS 564 Numerical Linear Algebra
CS 565 Numerical Methods for Optimization
CS 566 Numerical Data Analysis
CS 665-666 Advanced Numerical Analysis I, II

Possible topics for CS 665-666 are:

1. Minimization and Solution of Non-Linear Equations
2. Numerical Approximation and Mathematical Programming
3. Application of Operator Theory to Numerical Analysis

b) Automata Theory

This option is designed for those students who are interested in fundamental questions about the processing of data, but who are not committed to any particular mechanism, electronic, biological, or social, for doing the processing. Anyone entering this option should possess sufficient mathematical maturity to cope with pure mathematics course in logic, recursive function theory and abstract algebra; he should also have sufficient breadth of interest to be able to communicate with specialists in fields remote from computer sicence and discern the informational facets of their problems.

Courses available include:

MA 352 Computable Functions
CS 457 Logic Circuits
CS 531 Formal Languages
CS 546 Theory of Automata
CS 556 Translation of Programming Languages
MA 451 Introduction to Logic
MA 571-572 Logic
CS 558 Artificial Intelligence
CS 615 Topics in Formal Systems

c) Computing Systems Design

Students specializing in this area have two options: (1) programming systems, and (2) computer design. In each option, students are expected to gain an appreciation of the relationships between programming and machine architecture; i.e., software/hardware interfaces and tradeoffs. Students will study machine structure and assembly language programming for various machines in order to gain an understanding of the relationships and distinctions between computer design and programming systems.

The programming systems option includes topics such as: Machine Architecture, Assembly Language Programming, Design, Implementation and Use of Higher Level Languages, Translation of Programming Languages, Operating Systems, Data Structures, and On-Line Computing Systems. The computer design option includes topics such as: Logic Circuits, Logical Structure of Computers, Machine Architecture, Assembly Language, Operating Systems, Arithmetic Units, Memory Units, Input-Output, and Control and Interrupt Systems.

Courses Available include:

CS 400 Assembly Language and System Software
CS 401 Introduction to Programming Languages and Processors
CS 402 Hardware Structure and Software Design
CS 453 Assembly Language Programming
CS 457 Logic Circuits
CS 459 Logical Structure of Computers

CS 545 Data Structures

CS 553 Fundamental Concepts of Programming Languages

CS 555 Non-Numeric Techniques

CS 556 Translation of Programming Languages

CS 557 Operating Systems

CS 559 On-Line Computing Systems

CS 611 Topics in Computer Graphics

CS 612 Topics in Operating Systems

CS 613 Topics in Programming Languages

CS 614 Topics in Computer Systems

## Basic Master Curriculae in Areas of Study

### General Degree Requirements

The University's basic requirements are discussed in the Graduate School Bulletin. The Department requires a candidate to complete an approved course program of at least 30 semester hours (B average). Eighteen hours must be at the 500 level or above. Of the 30 hours, at least 24 will be in this Department, or selected from the suggested list of courses in other departments. Students will be expected to plan a program of study, in consultation with their departmental advisor, during the first semester of study.

The Master's Degree may be taken under Plan I (thesis required) or Plan II (non-thesis option). The thesis may count 4 to 6 credit hours. Students electing Plan II will read selected technical papers for three hours of independent study credit, and be required to pass an examination based on these papers. Appropriate papers will be listed by the Department at the beginning of each semester. In addition, every student (Plan I or Plan II) in the Master's program (beginning with the Fall Semester, 1973) will be required to pass a written Comprehensive Examination covering the fundamental aspects of Automata Theory, Computing Systems, and Numerical Analysis.

### Approved Electives

Courses not listed in the description of the fields of study may be appropriate for a given student's program. With the approval of an advisor, any courses are allowed under this heading.

## Computer Science as a Minor

Students majoring in another department may wish to take enough hours in computer science to satisfy the minor requirement of their major departments. Each such student will be assigned a faculty advisor from the Department of Computer Science, who will review his proposed plan of study for coherence and adequacy.

Attachment 4

# READING LISTS FOR
# PROGRAMMING AND SYSTEMS OPTION
# SPRING 1973

The student must select either one of the following two lists:

I. Operating Systems List

1. A. G. Fraser, "User Control in a Multi-Access System," Computer Journal, 11 (1968) p. 12.
2. P. J. Denning, "Virtual Memory," Computing Surveys, ACM, 2 (September 1970) pp. 153-191.
3. D. F. Hartley, B. Landy, and R. M. Needham, "The Structure of a Multiprogramming Supervisor," Computer Journal, 11 (1968) p. 247.
4. R. C. Holt, "Some Deadlock Properties of Computer Systems," Computing Surveys, ACM, 4 (September 1972) pp. 179-196.
5. G. J. Burnett and E. G. Coffman, "A Study of Interleaved Memory Systems," AFIPS Proceedings, 36 (1970 SJCC) p. 467.

II. Programming Languages List

1. Feldman and Gries, "Translator Writing Systems," CACM, 11 (February 1968) pp. 77-113.
2. Busam, "Dictionaries for Compilers," Data Structures in Programming Languages, ACM SIGPLAN Conference Proceedings, February 1971, p. 287.
3. D. J. Cohen and C. C. Gotlieb, "A List Structure Form of Grammars for Syntactic Analysis," Computing Surveys, ACM, 2 (March 1970) p. 65.
4. P. Branquart, J. Lewi, J. Heymans, and D. L. Parnas, "The Composition of Semantics in ALGOL 68," CACM, 14 (November 1971)
5. American National Standards Institute, FORTRAN, ANS X 3.9-1966.

# M. S. DEGREE READING LIST FOR SPRING 1973

## Numerical Analysis

1. C. Clenshaw and A. Curtis, "A method for numerical integration on an automatic computer," Numer. Math., Vol 2., pp. 197-205.

2. J. Oliver, "A doubly-adaptive Clenshaw-Curtis quadrature method," Comp. Journ., Vol. 15, No. 2, 1972, pp. 141-147.

3. W. G. Bickley, "Piecewise cubic interpolation and two-point boundary problems," Comp. Journ., Vol. 11 (1968) pp. 206-208.

4. D. J. Fyfe, "The use of cubic splines in the solution of two point boundary value problems," Comp. Journ., Vol. 12, 1969, pp. 188-192.

5. S. Conte, "The numerical solution of linear boundary value problems," SIAM Rev. 8 (1966), pp. 309-321.

6. J. Wilkinson, "Global convergence of Tridiagonal QR algorithms with origin shifts," Linear Algebra, Vol. 1, 1968, pp. 409-420.

7. G. Peters and J. Wilkinson, "The least squares problem and pseudo-inverses," Comp. Journ., Vol. 13, No. 2, 1970, pp. 309-317.

The student MUST read all of the above.

Comprehensive Exam Reading List In Automata Theory

DEPARTMENT OF COMPUTER SCIENCE - Spring 1973

1. Moore, Edward F. (1956), "Gedanken-Experiments on Sequential Machines," Automata Studies C.E. Shannon and J. McCarthy (Eds.), Princeton, 129-153.

2. Rabin, Michael O. and Scott, Dana (1959), "Finite Automata and Their Decision Problems," IBM Journal of Research and Development, 3, No. 2, 114-125 (April, 1959).

3. Eggan, L. C. (1963), "Transition Graphs and the Star-Height of Regular Events," Michigan Mathematical Journal 10, 385-397.

4. Scott, Dana, "Some Definitional Suggestions for Automata Theory," Journal of Computer and System Science 1 (1967), p. 187-212.

5. McCarthy, John (1963), "A Basis for a Mathematical Theory of Computation," (revised and extended version of McCarthy and Hirschberg (Eds.), North-Holland, Amsterdam, 33-70.

6. Y. Bar-Hillel, Perles, and Shamir, "On Formal Properties and Simple Phrase Structure Grammars," Language and Information, Y. Bar-Hillel, Chapter 9, Addision-Wesley, 1964.

The student must present all of the papers above

Attachment 5 8/13/73

# Master's Thesis Topics

Johnson, Sharon G., "The Floor Plan"

Botway, Lloyd F., "A Machine-Independent System for the Graphical Manipulation of Flow Charts"

Coleman, Samuel S., "Computer Concordance Making"

Donnelly, John L., "STUCOM--An ASA FORTRAN IV Student Compiler Written in FORTRAN"

Erickson, Elizabeth R., "A Contrastive Study and Simulation of Various Approaches to Macro Processing"

Frenkiel, Michel, "On the Conditioning of Linear Programming Problems"

Holzer, Charles B., "A Time Sharing, Paper Tape Editor for the Data General Nova"

Horowitz, Bernard, "A Processor-Independent, File Driven Computer-Assisted Instruction System"

Korgel, Clayton C., "Adaptive Signal Detection in Noise of Time Varying Power."

Lojko, Matthew, "Computer Implementation of Interval Arithmetic and Its Use in Gran-Schmidt Orthogonalization"

Mauro, Thomas, "Sharpened Analysis of Uniform Random Number Generators and the W-Test for Normality"

Snyder, David C., "The Design and Implementation of An On-Line Accouting System for Computer Facility Usage"

Working, Robert D., "An Outline for the Implementation of APL on a Small Computer

Campbell, Gary, "An Investigation of Markovian State Change Languages"

Kasik, David, "STUCOM--A Further Iteration on a Student Compiler"

Lang, Dorothy, "Style: An Exercise in Transforming FORTRAN Programs into a Stylized Text"

Meyer, Cynthia, "An Analysis of Computer Utility Feasibility"

McEwen, Julie, "An Application of Singular Value Decomposition to HO's Algorithm"

Paul, Malcolm, "An Extension to BASIC for Process Control"

Montgomery, James Francis, "Associative Memory--the Key to Improved Simscript"

Schwanke, Lee MacLaren, "MACS--A Programmable Pre-Processor with Macrogeneration

Scott, Sidney, "An Analysis of the Hasp Remote Batch Workstation Program"

Stevenson, Wilfred, "On the Automatic Generation of Optimal Lexical Analyzers"

Whitesell, William, "A System Designed for Improved R&D Budget Planning Through Use of Time-Sharing Computer Programs"

Wolf, Terry, "A Step Towards a Mathematical Predictor of the Iron Lung"

Alber, Chad, "Computer-Assisted Manuscript Preparation"

Baldwin, P. Thomas, "An Experimental Computer Model to Predict the Distribution of Highway-Associated Air Pollution"

Ourusoff, Nicholas, "An Interative Programmed-Instruction Driver, CAIPI"

Attachment 6 8/13/73

## Occupations of Students Who Have Graduated

| Students | Year | Occupation |
|---|---|---|
| Sharon G. Johnson (M.S.) | 1970 | Univ. of Colo. Computing Center |
| Lloyd F. Botway (M.S.) | 1970 | Anaconda Copper Limited |
| Samuel S. Coleman (M.S.) | 1970 | Working on Ph.D. in EE & an RA in EE |
| James T. Drummonds (M.S.) | 1970 | Unknown |
| John L. Donnelly (M.S.) | 1971 | Teaching at Metro State in Denver |
| Elizabeth R. Erickson (M.S.) | 1971 | Univ. of Colo. Computing Center |
| Rodney W. Firl (M.S.) | 1971 | U.S. Air Force |
| Michel Frenkiel (M.S.) | 1971 | French Army |
| Charles B. Holzer (M.S.) | 1971 | UNIVAC |
| Bernard Horowitz (M.S.) | 1971 | Bronx High School of Science |
| Clayton C. Korgel (M.S.) | 1971 | Unknown |
| Matthew Loiko (M.S.) | 1971 | NOAA |
| Thomas Maruo (M.S.) | 1971 | U.S. Army |
| David C. Snyder (M.S.) | 1971 | Proctor & Gamble Company |
| Robert D. Working (M.S.) | 1971 | Martin Marietta Corporation |
| Gary Campbell (M.S.) | 1972 | Hewlett Packard |
| David Kasik (M.S.) | 1972 | Batelle Corporation |
| Dorothy Lang (M.S.) | 1972 | Texas Instruments |
| William Mannion (M.S.) | 1972 | Unknown |
| Cynthia Meyer (M.S.) | 1972 | Systematics |
| Julie McEwen (M.S.) | 1972 | Upjohn |
| Malcolm Paul (M.S.) | 1972 | Consumers Power Company |
| James Montgomery (M.S.) | 1972 | U.S. Air Force |
| Lee MacLaren Schwanke (M.S.) | 1972 | Boeing Corporation |
| Sidney Scott | 1972 | Bell Laboratories |
| Wilfred Stevenson | 1972 | NOAA |
| Braxton Tabb (M.S.) | 1972 | Hughes Aircraft |
| William Whitesell | 1972 | Unknown |
| Terry Wolf (M.S.) | 1972 | Control Data Corporation |
| Chad Alber (M.S.) | 1973 | General Automation |
| Charles Bozeman (M.S.) | 1973 | U.S. Air Force |
| Linus Davis (M.S.) | 1973 | Martin Marietta |
| Warren Dellinger (M.S.) | 1973 | Texas Instruments |
| Vasumati Deshpande (M.S.) | 1973 | Housewife |
| Paul Gilmartin (M.S.) | 1973 | Unemployed |
| Fred Goldbaum (M.S.) | 1973 | Gates Rubber Company |
| Richard Orsborn (M.S.) | 1973 | Scientific Software Corporation |
| Irwin Zufelt (M.S.) | 1973 | IBM |
| P. Thomas Baldwin (M.S.) | 6/73 | Burroughs Corporation |
| Steven Bankes (M.S.) | 6/73 | Teaching Assistant, Comp. Sci. |
| Larry Ellis | 6/73 | East Texas State College |
| Jerry Heglar | 6/73 | IBM |
| Nicholas Ourusoff | 6/73 | Teaching Assistant, Comp. Sci. |
| H. Kent Weldon | 6/73 | Western Interstate Comm. for Higher Education |

| | | |
|---|---|---|
| Dennis A. Becker (M.S.) | 8/73* | |
| Ingrid Earn (M.S.) | 8/73* | Education Comm. of the States |
| Charles Hart (M.S.) | 8/73* | IBM |
| Stepan Maziar (M.S.) | 8/73* | |
| John H. Merrill (M.S.) | 8/73* | NCAR |
| Robert Mitchell (M.S.) | 8/73* | Xytex |
| John Penzien (M.S.) | 8/73* | Highway Administration |
| Larry Weber (M.S.) | 8/73* | IBM |
| Nancy White (M.S.) | 8/73* | IBM |
| Jeffrey Wright (M.S.) | 8/73* | |
| Jacob Wu (M.S.) | 8/73* | Brookhaven National Laboratory |

*Tentative

Attachment 7

4-25-73

# BUSINESS ORIENTED COMPUTING CURRICULUM

## ** DEPARTMENT OF COMPUTER SCIENCE **
## UNIVERSITY OF COLORADO

## I. Prerequisites

Bachelor's Degree in Business, with area of emphasis in Computer Based Information Systems; or the one year option of 24 hours of 400-level Business courses as listed under the General Requirements for the Master's Degree in the Bulletin of the Graduate School of Business Administration; or the equivalent of these options.

In addition, the following courses (or equivalent) will be required for admission as a regular degree student:

### A. <u>Programming</u>

| | |
|---|---|
| CS-202 | Intro to Computer Science for Business Majors (4 hours) |
| CS-312 | Business Data Processing Methods (3 hours) |
| CS-400 | Assembly Language and System Software (3 hours) |

### B. Mathematics

Nine hours of upper division mathematics, as outlined in the Master of Science Computer Science Degree Program description.

## II. Curriculum

### A. Recommended Courses

| | | |
|---|---|---|
| | CS-401 | Programming Languages and Processors (3 hours) |
| | CS-465 | Intermediate Numerical Analysis I (3 hours) |
| | CS-545 | Basic Data Structures (3 hours) |
| | CS-546 | Automata Theory (3 hours) |
| | CS-559 | On-Line Computing Systems (3 hours) |
| | CS-5XY | Data Management and File Systems (3 hours) |
| | M.SC445 | Information System Analysis (3 hours) |
| | CS-700 | Master's Thesis (6 hours) |
| or | CS-701 | Master's Reading Option (3 hours) |

### B. Recommended Options

| | |
|---|---|
| CS-402 | Computer Architecture (3 hours) |
| CS-616 | Topics in Data Processing (3 hours) |
| M.SC625 | Computer Oriented Decision Modeling (3 hours) |
| M.SC635 | Mathematical Programming (3 hours) |

A total of 30 credit hours at the 400 level or greater is required for graduation. Eighteen credit hours must be at the 500 level or greater. CS-700 or CS-701 must be taken. In addition, a written comprehensive exam must be passed.

Attachment 8

8/23/73

## Department of Computer Science Faculty Publications (1970-1973)

### SUBMITTED, ACCEPTED, AND PUBLISHED

Bailey, Daniel

"Cluster Analysis of a Scientific Field," Proceedings of the General Systems Society, 1972.

"The Computer Laboratory for Instruction in Psychological Research," CLIPR, Beha . Res. Methods and Instrumentation, 1972

Probability and Statistics: Models for Research. New York: Wiley, 1971.

Ehrenfeucht, Andrzej

"Practical Decidability," Journal of Symbol. Logic, February 1970.

"A Finitely Axiomatizable Complete Theory with Atomless $F_1$ (T)," with Fuhrken, G., Arch. Math. Logik, Grundlagenforsch, 1970.

"On Models with Undefinable Elements," with Fuhrken, G., Math. Scand., 1970.

"Existence of $2^{X}{}_{X}$ Categorical Theories," with Glassmire, Wm., Bul. Pol. Acad. of Science.

"A Criterion for a Finitely Generated Group to be Finite (An Application of Markov Processes)," Ind. Math. 33, 1971.

"Abbreviating Proofs by Adding New Axioms," with Jan Mycielski, BAMS 77, No. 3, 1971.

"Slender Algebras," with Fajtlowski, S., and Mycielski, J., AMS

"Do Nilpotent Groups Always Have Equipotent Abilian Subgroups?," with Faber, V., Ind. Math

"Interpolation of Functions Over Measure Spaces," with Mycielski, J., Jour. of Appr. Theory.

"Interpolation of Functions Over a Measure Space and Conjectures about Memory," with J. Mycielski, Procedings of National Academy of Sciences, Biophysiological Series.

"A Semantics for Non-iterated Local Observations (Part I), with Geiser, J., Gordan, C. E., and deJongh, D.H.J., preprint.

Ellis, Clarence

"The Halting Problem for Probabalistic Context-Free Generators," Journal of the Association for Computing Machinery, Vol. 19, No. 3, July 1972, pp. 369-399.

"Probabalistic Tree Automata," Information and Control Magazine, Vol. 19, No. 5, December 1971, pp. 401-416.

"Generalizations of Macro Processor Systems," Taiwan Association of Computer Science Journal, April 1972, pp. 9-15.

"Parallel Compiling Techniques," Proceedings of the 1971 ACM National Conference, pp. 508-519.

Fairley, Richard

"Batch to Interactive Conversion of Digital Simulation Languages," 1971 IEE Region Six Conference Proceedings, May 1971.

"Batch and Interactive Simulation of Partial Differential Equation Models," 1971 Summer Computer Simulation Conference, July 1971.

"A Batch-Compatible Interactive Graphics Software System for Dynamic Modelling," Ph.D. Dissertation, UCLA, July 1971.

Fosdick, Lloyd D.

"The Production of Better Mathematical Software," Comm. ACM 15 (1972), pp. 611-617.

Gabow, Harold

"An Efficient Implementation of Edmonds' Maximum Matching Algorithm, Stanford University Technical Report, No. 328, June 1972.

"Implementation of Algorithms for Maximum Matching on Nonbipartite Graphs, Stanford University, Ph.D. Thesis, August 1973.

Gary, John

"A Matrix Method for Ordinary Differential Eigenvalue Problems," with R. Helgason, Journal of Computational Physics, Vol. 5, No. 2, 1970, pp. 169-189.

"An Extension of FORTRAN Containing Finite Difference Operators," with R. Helgason, Software-Practice and Experience, 2, 1972, pp. 321-336.

"Estimate of Truncation Error in Transformed Coordinate, Primitive Equation Atmospheric Models," Journ. Atmos. Sci. 30, March 1973, pp. 223-233.

Kaufman, Linda

"Function Minimization and Automatic Therapeutic Control," Stanford Computer Science Report 228, July 1971.

"A Generalization of the LR Algorithm to Solve $Ax=\lambda Bx$," Stanford Computer Science Report 276, April 1972.

Krueger, E. Rex

"The Use of Interactive Computer Graphics in Soil Engineering Analyses and Design," with R. L. Schiffman and J. R. Stein, Proceedings, Computer Graphics 70, Second International Symposium, Brunel University, Uxbridge, England, April 1970.

"Final Report: Seminar for Directors of Academic Computing Centers," (Editor), August 1970.

"Proceedings, Second Annual Seminar for Directors of Academic Computer Centers," (Editor), August 1971.

"Computer Use in Civil Engineering," with R. L. Schiffman, Engineering Issues - Journal of Professional Activities, ASEE, Vol. 98, pp. 251-273, April 1972.

"On the Stability of Viscous Flow Between Two Rotating Noncentric Cylinders," with R. L. Urban, Journal of the Franklin Institute, Vol. 293, No. 3, pp. 153-172, March 1972.

Maybee, John

"External and Materially Damped Three Dimensional Rotor Systems," with J. Genin, J. of Nonlinear Mechanics, Vol. 5, 1970, pp. 287-297.

"Whirling Motion of a Viscoelastic Continuous Shaft," with J. Genin, International J. Engr. Science, Vol. 8, 1970, pp. 674-686.

"Remarks on the Follower Problem and the Euler Method in Statis Stability," with J. Genin, ZAMP, Vol. 22, pp. 1177-1182, 1971.

"Boundedness Theorem for a Nonlinear Mathieu Equation," with J. Genin, Quart. Appl. Math, 1970, pp. 450-453.

"Nonconservative Linear Systems with Constant Coefficients," with J. Genin, J. Inst., Math Appl's, 1971, pp. 358-370.

"On the Asymptotic Stability of Solutions of a Linear Viscoelastic Beam," with J. Genin, Journal of the Franklyn Institute, Vol. 293, 1972, pp. 191-197.

"Further Remarks on the Theory of Three Dimensional Whirling," with J. Genin, J. Nonlinear Mech., Vol. 6, 1971, pp. 663-667.

"Role of Material Damping in the Stability of Rotating Systems," with J. Genin, J. Sound and Vibrations, 1971, pp. 399-404.

"Introduction to Applied Mathematics, Vol. 1," with J. Genin, January 1970, Holt, Rinehart, and Winston, Inc.

"Manuscript in Progress of Monograph in Qualitative Problems in Matrix Theory," with J. Quirk. This was requested by Richard Bellman for his series of monographs.

Nutt, Gary

"The Boeing Dayfile Analysis Modified for the University of Washington Computer Center" University of Washington Computer Center Technical Report No. 4, October 1970.

"SIM6000 Program Description and Use," University of Washington Computer Science Group Technical Report No. 71-07-05, July 1971.

"Validation of a Trace-Driven CDC 6400 Simulation," with J. D. Noe, 1972 Spring Joint Computer Conference.

"Some Evaluation Net Macro Structures," with J. D. Noe, University of Washington Computer Science Group Technical Report, No. 73-01-07, January 1973.

"Multiprogramming and Multiprocessing System Descriptions," with J. D. Noe, University of Washington Computer Science Group, Technical Report No. 73-01-08, January 1973.

"Evaluation Nets for Computer System Performance Analysis," Proceedings of the AFIPS Fall Joint Computer Conference, Vol. 41, Pt. I, pp. 279-286, December 1972.

"The Formulation and Application of Evaluation Nets," National Technical Information Service, Accession No. PB-214 858, 169 pages, February 1973.

"The Computer System Representation Problem," Proceedings of Symposium on the Simulation of Computer Systems, June, 1973.

"Macro E-nets for Representation of Parallel Systems," with J. D. Noe, IEEE Transactions on Computers, Vol. C-22, No. 8, pp. 718-727, August, 1973.

Osterweil, Leon

"Some Algorithms in Graph Enumeration," Doctoral Dissertation, University of Maryland, 1971.

"A Theorem for Enumerating Certain Types of Collections," Can. J. Math., Vol. XXV, No. 1, 1973, pp. 74-82.

Poole, Peter

"Developing a Multi-Access System On Line," Software-Practice and Experience, 1, 39-50, 1971.

"Hierarchial Abstract Machines," Culham Laboratory Software Engineering Symposium, April 1971.

"Software Portability," Proceedings of Software Conference, July 1971, p. 32.

"Abstract Machine Modelling to Produce Portable Software," Software-Practice and Experience, 1972.

## SUBMITTED, ACCEPTED, NOT YET PUBLISHED

Ellis, Clarence

"On the Probability of Deadlock in Computer Systems," accepted for presentation at the 4th National ACM Operating Systems Conference, October 15-17, 1973.

"Non-Markov Models of Computer Deadlock," accepted for presentation at the 1973 Joint SIAM-IMS Conference on Probability and Statistics, October 8-10, 1973.

Maybee, John

"Mechanical Vibrations Trees," forthcoming in J. Math. Analysis and Applications.

"Combinatorially Symmetric Matrices," forthcoming in J. Linear Algebra.

Osterweil, Leon

"Some Classes of Uniquely Colorable Graphs," forthcoming in Discrete Mathematics.

## SUBMITTED, PENDING

Ehrenfeucht, Andrzej

"Decidable Theories Without Finitely Axiomatizable Complete Extensions," submitted to J.S.L.

"An Algorithm for Finding the Elementary Circuits of a Directed Graph," with Lloyd D. Fosdick and Leon J. Osterweil, SIAM Journal of Computing, submitted August 1973.

Ellis, Clarence

"Approximations of Probabalistic Turing Machines by Probabalistic Pushdown Automata," submitted to the Journal of Computer and Systems Sciences, September 1972.

"Computer Science Education in Taiwan--An AFIPS Internship Report," submitted to AFIPS and to the CACM, October 1972.

Fairley, Richard

"Continuous System Simulation Languages: Design Principles and Implementation Techniques," submitted to SIGPLAN Symposium on Very High Level Languages.

Fosdick, Lloyd

"An Algorithm for Finding the Elementary Circuits of a Directed Graph," with Andrzej Ehrenfeucht and Leon Osterweil, SIAM Journal of Computing, submitted August 1973.

Gabow, Harold

"An Efficient Implementation of Edmonds' Maximum Matching Algorithm," submitted for publication.

Gary, John

"A FORTRAN Extension for Data Display," submitted to IEEE Transactions on Computers.

"Compact Representation of Contour Plots for Phone Line Transmission," with J. Adams, submitted to Comm. ACM.

Book - Numerical Solution of Partial Differential Equations.

Maybee, John

"The GM-Matrix Problem," with J. Quirk.

Nutt, Gary

"Some Applications of Finite State Automata Theory to the Deadlock Problem," submitted to the Comm. ACM, April 1973.

Osterweil, Leon

"A Stochastic Model of Multiprocessor Access to an Interleaved Memory," Comm. ACM, submitted November 1972.

"An Algorithm for Finding the Elementary Circuits of a Directed Graph," with Lloyd D. Fosdick, and Andrzej Ehrenfeucht, SIAM Journal of Computing, submitted August 1973.

Zeiger, H. Paul

"Approximate Linear Realizations via Ho's Algorithm," with McEwen, submitted to IEEE Trans. on Automatic Control.

# UNIVERSITY OF COLORADO

## DEPARTMENT OF COMPUTER SCIENCE

## Technical Reports

"The Production of Better Mathematical Software" by Lloyd D. Fosdick
#CU-CS-001-72

"A Theorem for Enumerating Certain Types of Collections" by
Leon Osterweil
#CU-CS-002-72

"Some Classes of Uniquely Colorable Graphs" by Leon Osterweil
#CU-CS-003-72

"Proceedings of the Conference on the Validation and Distribution of
Computer Software" by Lloyd D. Fosdick
#CU-CS-004-72

"Enumeration of Non-Separable Graphs on Fewer Than Ten Points"
Leon Osterweil
#CU-CS-005-72

"A Stochastic Model of Multiprocessor Access to an Interleaved Memory,"
by Leon J. Osterweil
#CU-CS-006-72

"SYTLE Editor: User's Guide," by Dorothy E. Lang
#CU-CS-007-72

"Practical Decidability," by Andrzej Ehrenfeucht
#CU-CS-008-72

"Interpolation of Functions Over a Measure Space and Conjectures About
Memory," by Andrzej Ehrenfeucht and Jan Mycielski
#CU-CS-009-72

"The Formal Definition of a Parameter Passing Language," by Richard E. Fairley
#CU-CS-010-72

"On Ulam's Problem," by Richard Dunn
#CU-CS-011-73

"Approximate Linear Realizations of Given Dimension Via Ho's Algorithm,"
by H. Paul Zeiger
#CU-CS-012-73

"Computer System Monitoring Techniques," by Gary J. Nutt
#CU-CS-013-73

"On the Organization of Memory" by Andrzej Ehrenfeucht and Jan Mycielski
#CU-CS-014-73

"Subwords in Deterministic Tol Languages" by A. Ehrenfeucht and G. Rosenberg
#CU-CS-015-73

"Semantic Models of Parameter Passing" by Richard E. Fairley
#CU-CS-016-73

"Some Applications of Finite State Automata Theory to the Deadlock Problem" by Gary J. Nutt
#CU-CS-017-73

"Some Etol Languages Which are not Deterministic" by A. Ehrenfeucht and G. Rozenberg
#CU-CS-018-73

"Computer Science Education in Taiwan - An Afips Internship Report" by Clarence A. Ellis
#CU-CS-019-73

"The Approximation of Probabilistic Turing Automata by Probabilistic Pushdown Automata" by Clarence A. Ellis
#CU-CS-020-73

"Combinatorially Symmetric Matrices" by John S. Maybee
#CU-CS-021-73

"The GM-Matrices Problem" by John S. Maybee
#CU-CS-022-73

"On the Structure of Polynomially Bounded Dol Systems" by A. Ehrenfeucht and G. Rozenberg
#CU-CS-023-73

"An Algorithm for Finding the Elementary Circuits of a Directed Graph" by L. Fosdick, L. Osterweil, and A. Ehrenfeucht
#CU-CS-024-73

"Codings of OL Languages" by A. Ehrenfeucht and G. Rozenberg
#CU-CS-025-73

"On the Probability of Deadlock in Computer Sciences" by Clarence A. Ellis
#CU-CS-026-73

"Nonterminals Versus Homomorphisms in Defining Languages for Some Classes of Rewriting Systems" by A. Ehrenfeucht and G. Rozenberg
#CU-CS-027-73

"A Characterization Theorem for a Subclass of ETOL Languages" by A. Ehrenfeucht and G. Rozenberg
#CU-CS-028-73

"Mechanical Vibration Trees" by Joseph Genin and John S. Maybee
#CU-CS-029-73

"The Number of Occurrences of Letters Versus Their Distribution in EOL Languages" by A. Ehrenfeucht and G. Rozenberg
#CU-CS-030-73

Attachment 10

8/13/73

## Department of Computer Science
## Grants

| INVESTIGATOR | TITLE | SUBMITTED TO | AMOUNT |
|---|---|---|---|
| Bailey, Daniel E. | Summer Institute for Computer Science and Social and Behavioral Science Education<br>January 1, 1973 - September 30, 1973 | NSF | $ 82,000 |
| Fairley, Richard | An Interactive System for Studying Semantic Models of Computer Programs<br>September 1, 1973 - September 1, 1975 | NSF | 90,365 |
| Fosdick, Lloyd D.<br>Osterweil, Leon J. | Research in the Analysis of Mathematical Software for the Purpose of Validation and Documentation<br>June 1, 1973 - November 30, 1975 | NSF | 236,400 |
| Fosdick, Lloyd D. | Support of a Computer Science Program<br>August 21, 1970 - August 20, 1973 | NSF | 285,000 |
| Fosdick, Lloyd D.<br>Cowell, Wayne | Planning an Approach to Testing and Dissemination of Computer Programs for Research and Development<br>November 15, 1971 - April 30, 1972 | NSF | 26,925 |
| Fosdick, Lloyd D. | Meeting on Validation and Distribution of Computer Software<br>November 15, 1971 - August 31, 1973 | NSF | 8,435 |
| Fosdick, Lloyd D. | Support for ACM Secretary<br>Starting July 1, 1972 | ACM | 3,600 |
| Maybee, John S.<br>Quirk, James | Qualitative Methods in Matrix Theory and their Applications<br>June 1, 1973 - June 1, 1974 | NSF | 7,500 |
| Osterweil, Leon J. | A Regional Distribution Center for the CITY Model (1 year grant) | EPA | 5,275 |
| Waite, William M. | Software Engineering: Portability and Adaptability (Continuation of GJ 32471)<br>March 1, 1974 - February 29, 1976 | NSF | 249,210 |

# Department of Computer Science
## Grants
## (Pending)

| INVESTIGATOR | TITLE | SUBMITTED TO |
|---|---|---|
| Maybee, John S.<br>Quirk, James | The Role of Qualitative Techniques in Economic Theory | NSF |
| Nutt, Gary J.<br>Ellis, Clarence A. | Multiprocessor Computer Configuration Study | Air Force Of. of Sci. Res; Office of Naval Res. |

Attachment 11

8/13/73

## Courtesy Faculty Members

The following members of the faculty of the University of Colorado have interests closely related to those of the Department and participate in our activities:

John F. Chaney, Vice Provost for Management Systems; B.S., University of Oklahoma. Administrative Data Processing.

J. Daniel Couger, Professor; D.B.A., University of Colorado. Design of Computer Based Management Information Systems, Design of Computerized Planning Models, and Cost/Effectiveness Analysis.

Harry F. Jordan, Assistant Professor; Ph.D., University of Illinois. Hardware-Software Interface, Computer Graphics and Pseudo-random Sequences.

E. Rex Krueger, Associate Professor, Ph.D., Rennselaer Polytechnic Institute. Computer Graphics.

Claude McMillan, Professor; Ph.D., Ohio State University. Hardware-software systems for solving large scale optimization problems, Large file management with emphasis on set theoretic data structures for management information systems, man-machine interaction in large scale systems simulation.

Robert Richtmyer, Professor; Ph.D., Massachusetts Institute of Technology. Numerical Methods, Scientific Uses of Computers.

Roland A. Sweet, Assistant Professor, Ph.D., Purdue University. Numerical Solutions of Partial Differential Solutions and Numerical Linear Algebra.

Stanislaw M. Ulam, Professor; D.Sc., Polytechnic Institute, Lwow (Poland). Application of Computer Simulation to Biology, Theory of Intelligence, and Biological Control Mechanisms.

William M. Waite, Associate Professor in Electrical Engineering; Ph.D., Columbia University. Software Engineering, Portability Software, Macroprocessors, Compilers, and Non-numeric Applications.

Attachment 12

8/13/73

## EMPLOYED STUDENTS
## as of Summer 1973

Mark Bailey, CU Computer Center
Steven Bankes, TA, Computer Science (Fall 1973)
Dennis Becker, CU Computing Center
Anthony Brittain, NOAA
Robert Brown, Univ. of Col. LASP
Mark Brummel, Mountain Bell Telephone
Joe Bryant, Universal Robotics and Cosmological Tool
Mark Chisolm, RA, Psychology
Allen Culnan, JILA
Fred Day, Bell Telephone Labs
Joseph Demic, CU Computing Center
Thomas Dodge, IBM
Herman Durrett, Inst. for the Study of Intellectual Behavior
Ingrid Earn, CU Computing Center
Patricia Farmer, Fellowship
Maryann Gaug, TA, Computer Science (Fall 1973)
Gilbert Green, NCAR
James Hallock, Community College of Denver
Paul Harrold, Harrold's Carpets, Inc.
Charles Hart, IBM
Jordan Hastings, NCAR
Harry Joel Jeffrey, Fellowship
Jerome Johnson, WICHE
John Kalk, TA, Psychology
Virginia Kallal, NOAA
William Kemper, TA, Computer Science (Fall 1973)
Michael Lane, Bell Telephone Labs
John Lybarger, Air Force Accounting and Finance Center
Nancy Martin, NCAR
Ronald Mayka, Bell Telephone Laboratories
Wayne Meitzler, Auto-trol
John Merrill, NCAR
JoAnne Miller, Inst. for the Study of Intellectual Behavior
Robert Mitchell, XYTEX
Janis Osterweil, CLIPR, Psychology
Nicholas Ourusoff, TA, Computer Science (Fall 1973)
John Penzien, Highway Administration
Brian Platte, IBM
Loretta Rainish, TA, Computer Science (Fall 1973)
Ralph Redman, Southern Colorado State College
James Reid, NROTC Unit, Univ. of Colorado
William Roberts, National Bureau of Standards
Buzz Ross, MCD Biology
Richard Sato, NCAR
Waldean Schulz, NDEA Fellowship
Loretta Shenk, IBM
Joseph Slansky, Boulder Valley Public Schools
David Smith, TA, Computer Science (Fall 1973)
Margaret Snyder, NOAA
Thomas Stuebe, Martin Marietta
Karl Sutterfield, NOAA
Jesse Tarshis, Mt. Bell Telephone
Larry Weber, ITS/OT Dept. of Commerce
Carol Weissmann, Community College
Herbert Weldon, WICHE
Nancy Munroe White, IBM
Richard White, Bell Labs
Stephen Winograd, TA, Computer Science (Fall 1973)

CU Proposal No. 73.7.337

Attachment 10

MEMORANDUM OF AGREEMENT BETWEEN THE
UNIVERSITY OF COLORADO AND THE CALCULTOR
DIVISION OF THE HEWLETT-PACKARD COMPANY
AT LOVELAND, COLORADO

SUBJECT: INTERNSHIP PROGRAM FOR COMPUTER SCIENCE STUDENTS.

PERIOD OF AGREEMENT: FROM AUGUST, 1973 TO JUNE, 1974.

Lloyd D. Fosdick July 16, 1973

Lloyd D. Fosdick, Chairman Date
Department of Computer Science

Lawson Crowe

Lawson Crowe, Provost, Date
Vice President for Research

Terry L. Gildea 10 Aug 73

Terry L. Gildea Date
Product Support Manager
Hewlett-Packard

Thomas L. Kelley Date
General Manager
Calculator Products Division
Hewlett-Packard

4. Supervision

The intern will be supervised by an employee of the Company and a member of the faculty of the Department of Computer Science. The award of a grade in CS-590 will be made by the faculty member on the basis of the final report submitted by the student and a work performance report submitted by the Company supervisor.

5. Selection

Selection of the intern will be made by the Company from a list of candidates supplied by the Department of Computer Science. The Company will be permitted access to the candidates' supporting documents required for admission to the Department: Transcripts, GRE scores, and letters of recommendation.

6. Stipend

As an employee of the University, the intern will receive a stipend for his work at the Company during the school year from September through June equal to that awarded research assistants (currently $3,200 per academic year), supplemented by a cost of travel allowance of $300.

7. Financing of the Stipend

The stipend will be paid by the Company on a monthly basis upon receipt of a statement from the Finance Office of the University of Colorado. A 3% accounting charge will also be paid by the Company. Payments will be made into an account established by the Finance Office for the Department of Computer Science to be designated the "Internship Account."