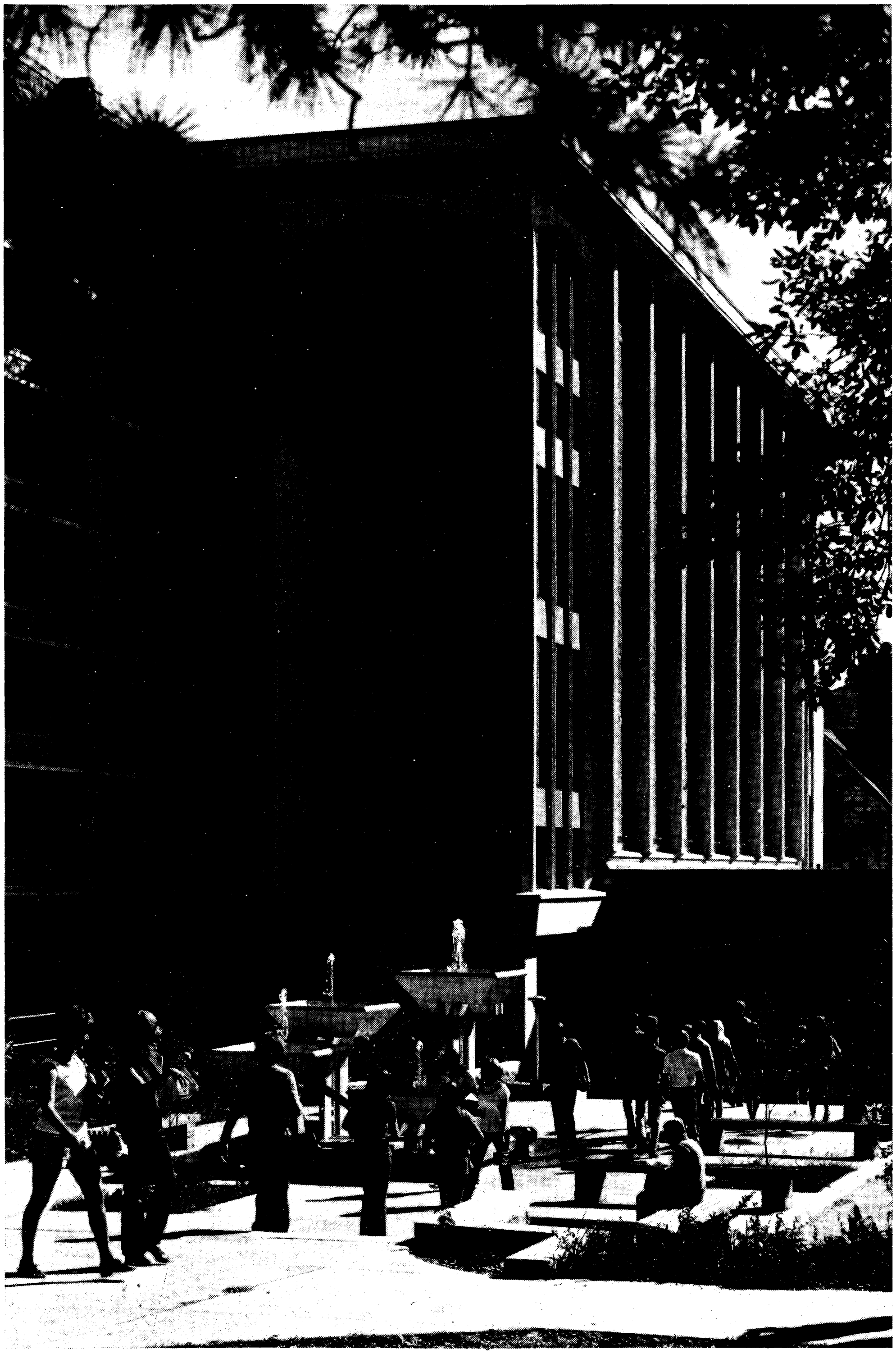
The Ph.D. Program in Computer Science at the University of Kansas and Kansas State University

The Ph.D. Program in Computer Science at the University of Kansas and Kansas State University



I. INTRODUCTION

Computer science has rapidly become an integral part of the higher educational pattern, equipping aspiring scientists, scholars, engineers, administrators, and teachers with the computer science understanding essential to their future careers. In response to the broad need for advanced training in computer science the University of Kansas and Kansas State University jointly offer the Ph.D. in this field. The doctor of philosophy degree in computer science is granted by each institution. The resources of both institutions are available to each doctoral candidate and his degree is granted by the institution with which his dissertation director is affiliated.

Application for admission to the Ph.D. program may be made to the Department of Computer Science at either the University of Kansas or Kansas State University. Once admitted to the program, a student in good standing need make no further application to enroll in classes on either campus.

Assistantships and fellowships are ordinarily available on each campus and are awarded according to the established procedures of each school and department.

II. REQUIREMENTS FOR THE PH.D. DEGREE IN COMPUTER SCIENCE

Each doctoral candidate will be required to take a candidacy examination, write a dissertation, and defend the dissertation in a final oral examination.

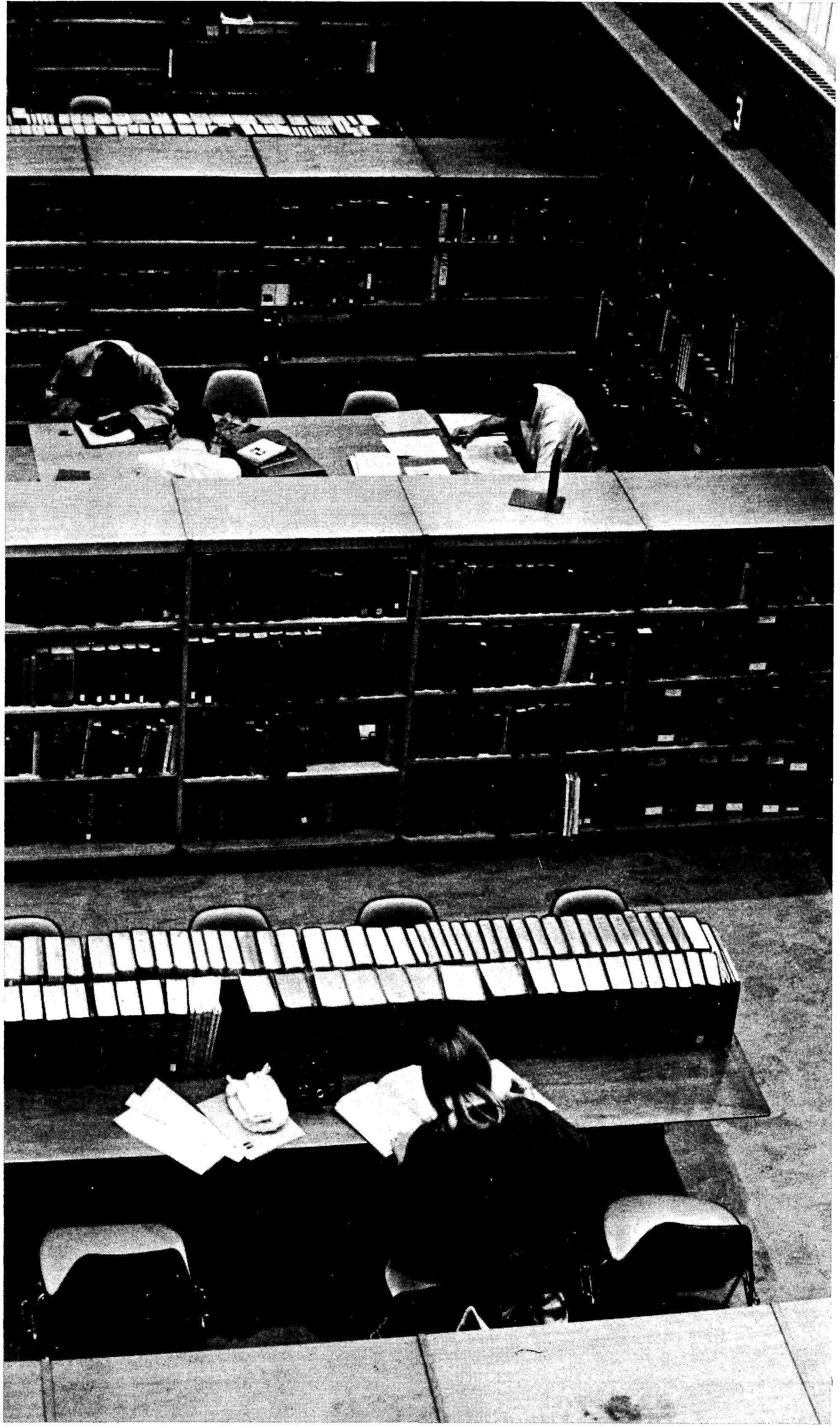
For initial admission to the Ph.D. program, the candidate files a "Declaration of Intent" (see last page of brochure) with the graduate studies committee in the appropriate department. Candidates who are simultaneously seeking admission to the institution and the Ph.D. program would normally include the "Declaration of Intent" with other admissions materials.

The candidacy examination will consist of both a written and an oral examination. It will be individually tailored to the student, and will be prepared and administered by the student's committee, formed when the student is admitted to the Ph.D. program. The student must have completed the major portion of his course work before the candidacy examination will be scheduled.

Passing the candidacy examination explicitly defines admission to candidacy. At that time, a dissertation committee is appointed. Normally, the dissertation committee is a three-member subcommittee of the student's committee with the student's research adviser as chairman. If the student wishes, faculty from both campuses may be on the dissertation committee, but this is not required.

When the completed dissertation has been submitted to the committee, a final oral examination will be scheduled. This examination will be announced to the Computer Science Department and other relevant departments on both campuses so that, in addition to the final examination committee, other interested faculty and students may attend.

The doctoral candidate will also be responsible for meeting any other general requirements (e.g., foreign language or skills requirement) at his institution.



Courses of Instruction

III. COURSES OF INSTRUCTION

The following courses are listed as they appear in the catalogs of the two institutions. Conventions of course numbering, prerequisites, and other special information contained in the lists are consistent with the individual institution's practice.

AT THE UNIVERSITY OF KANSAS

Faculty: Bavel, Boggs, Bulgren, Case, Hetherington, Horowitz, Mansfield, Schweppe, S. Sedelow, W. Sedelow, Tuggle.

FOR UNDERGRADUATE CREDIT ONLY

16. Introduction to Computing. (2) Algorithms, programs, and computers. Expressions, statements, and program structure. Programming and computing systems. Debugging and verification of programs. Data representation. Organization and characteristics of computers. Survey of computers, languages, computing systems and applications. Computer solution of several problems using various languages. Prerequisite: Mathematics 2a, or 2c, or 10, or 11, or equivalent.

FOR JUNIORS, SENIORS AND GRADUATE STUDENTS

Note: Courses C.S. 100, C.S. 110, and C.S. 120 may not be taken for graduate credit by computer science majors.

100. Computers and Programming. (3) Computer structure, machine language, instruction execution, addressing techniques, and digital representation of data. Computer systems organization, logic design, micro-programming and interpreters. Symbolic coding and assembly systems, macro definition and generation, and program segmentation and linkage. Systems and utility programs, programming techniques, and recent developments in computing. Several computer projects to illustrate basic machine structure and programming techniques. Prerequisite: Course 16, or equivalent.

110. Logic, Algorithms, and Graph Theory. (3) Review of sets, relations and mappings. Boolean algebra and propositional logic. Algorithms and abstract machines. Elements of graph and network theory. Applications to the computer and information sciences. Prerequisites: Course 16 and Mathematics 11 or 22, or equivalent.

120. Fundamentals of Symbol Processing. (3) An introduction to computer-based symbol manipulation, including data representation, algorithm development, programming languages, computer organization, and input-output options. Computer analysis of languages, of other communication modes such as music and art, of human artifacts, and of behavioral phenomena. Solution of several basic information processing problems using the computer.

132. Pattern Recognition and Pattern Generation. (3) Computer-based techniques for inputting, manipulating and analyzing, and outputting the aural, visual and tactile data of the fine arts, humanities, and social sciences. Computer-generation of art, music, and literary forms. Analysis of artifacts. Encoding, software, hardware, graphics. Prerequisite: Course 16 or 120.

150. Data Structures. (3) Basic concepts of data. Linear lists, strings, arrays, and orthogonal lists. Representation of trees and graphs. Storage systems and structures, and storage allocation and collection. Multilinked structures. Symbol tables and searching techniques. Sorting (ordering) techniques. Formal specification of data structures, data structures in programming languages, and generalized data management systems. Prerequisites: Courses 100 and 110, or equivalent.

160. Programming Languages. (3) Formal definition of programming languages including specification of syntax and semantics. Simple statements including precedence, infix, prefix, and postfix notation. Global properities of algorithmic languages including scope of declaration, storage allocation, grouping of statements, binding time of constituents, subroutines, coroutines, and tasks. Run-time representation of program and data structures. Prerequisites: Courses 100 and 110, or equivalent.

170. Computer Organization. (3) Basic digital circuits, Boolean algebra and combinational logic, data representation and transfer, and digital arithmetic. Digital storage and accessing, control functions, input-output facilities, system organization, and reliability. Description and simulation techniques. Features needed for multi-programming, multi-processing, and real-time systems. Other advanced topics and alternate organizations. Prerequisites: Courses 100 and 110, or equivalent.

180. Numerical Calculus. (3) An introduction to the numerical algorithms fundamental to scientific computer work. Includes elementary discussion of error, polynomial interpolation, quadrature, linear systems of equations, solution of nonlinear equations, and numerical solution of ordinary differential equations. The algorithmic approach and the efficient use of the computer are emphasized. Prerequisites: Course 16 and Mathematics 55, or equivalent.

181-182. Numerical Analysis I and II. (3,3) (Same as Mathematics 181-182)

190. Special Topics. (1-3) Arranged as needed to present appropriate material to groups of students. May be repeated for additional credit. Prerequisite: Variable.

192. History of Computing Technology and Information Science. (3) Important ideas, inventions, and leaders from the seventeenth century to the present. Particular attention to the sources of innovations, the uses made of computer science knowledge, and the development of a separate academic discipline. Prerequisite: Course 16 or 120.

194. Social Issues in Computer Science. (3) A critical examination of attempts to assess the societal consequences of current, prospective, and proposed uses of computers. The present status of computer science as a profession. Contemporary situation of computer science professionals vis-a-vis industry, government, education, the professions, and the general public. Comparative analysis of computer science in other countries. Prerequisite: Course 16 or 120.

199. Directed Reading. (1-3) Reading under the supervision of an instructor on a topic chosen by the student with the advice of an instructor. May be repeated for additional credit. Consent of the department required for enrollment. Prerequisite: Instructor's permission.

The Ph.D. Program
in Computer Science
at
Kansas State University
and the
University of Kansas

FOR SENIORS AND GRADUATE STUDENTS

200. Systems Programming. (3) Review of batch process systems programs. Implementation techniques for parallel processing of input-output and interrupt handling. Addressing techniques, core management, file system design and management, system accounting, and other user-related services. Overall design of operating systems. Multiprogramming systems on multi-processor hardware. Traffic control, interprocess communication, design of system modules, and interfaces. System updating, documentation and operation. Prerequisites: Courses 150, 160, and 170, or equivalent.

210. Introduction to Automata Theory. (3) Structure, decompositions, mappings, and applications of sequential machines and other automata. Prerequisites: Course 110 or 211, and Mathematics 158; or equivalent.

211. Applied Boolean Algebra. (3) Boolean algebra with applications to switching circuits, sets, and logic. Prerequisite: Math 11 or 22, or equivalent.

212. Mathematical Logic. (3) Propositional calculus. First order theories and model theory. Elementary arithmetic and Godel's incompleteness theorems. Prerequisite: Math 165 or 191, or equivalent evidence of mathematical maturity. (Same as Mathematics 212).

213. Recursive Function Theory. (3) Effective computability and Church's thesis, unsolvability results, recursively enumerable sets, reducibilities, finite and infinite extension arguments, applications, priority arguments, and hierarchy theory. Prerequisite: Course 212. (Same as Mathematics 213).

216. Formal Language Theory. (3) Representation, classification, properties and grammars of formal languages. The relation of formal languages to automata. Decidability questions. Prerequisite: Course 210 (may be taken concurrently, with permission of the instructor).

220. The Computer as Instrumentation for Research in the Humanities and Social Sciences. (3) Topics and problems chosen in relation to the research interests of the seminar members, with special attention to new applications possibilities relevant for social scientists and humanists. Prerequisite: Course 120 or permission of instructor.

223. Computational Linguistics. (3) Preparation of dictionaries and concordances; distribution studies; parsing algorithms for context-free, context-sensitive, and transformational grammars. Prerequisites: Course 16 or 120 and Linguistics 101. (Same as Linguistics 283.)

225. Computational Stylistics. (3) Computational approaches to the study of style in language and literature. Stylistic discrimination by culture, genre, and author, with attention to problems in stylistic development. Survey and use of computer program packages. Prerequisite: Course 16 or 120, or equivalent. (Same as Linguistics 285).

230. Simulation of Cognitive Processes. (3) An introduction to "creative" information processing systems including examples of individual and coordinated human behavior, and artificially intelligent computer programs. Elementary systems analysis, simulation techniques and heuristic programming as needed to study cognitive processes. Construction and validation of a simulation of some "intelligent" information-processing system. Prerequisites: Courses 150 and 160; Course 235 also desirable.

235. Systems Simulation. (3) Introduction to Monte Carlo methods and their applications. Random variable generation, queuing, and variance reducing techniques. Definition and use of special languages for discrete simulation. Selected examples of simulations in business, engineering, industry and science. Prerequisites: Course 100 and Mathematics 127, or consent of instructor.

255. Information Retrieval. (3) Information analysis, dictionary construction, and automatic information systems. Dictionary, statistical and syntactic operations. Retrieval models and processes. Input-output systems including auxiliary services. Evaluation of retrieval systems. Prerequisite: Course 150.

260. Programming Structures. (3) Relation of programming structures to data and machine structures, static and dynamic structures, recursive and iterative structures, program and command structures, microprogram and macroprogram structures, and extension of language and data structures. Prerequisite: Course 200.

265. Compiler Construction. (3) Review of program language structures, translation, loading, execution, and storage allocation. Compilation of simple expressions and statements. Organization of a compiler including compile-time and run-time symbol tables, lexical scan, syntax scan, object code generation, error diagnostics, object code optimization techniques, and overall design. Use of compiler writing languages and bootstrapping. Prerequisites: Courses 150 and 160, or equivalent.

270. Computer System Design. (3) Study of such problems as arithmetic and non-arithmetic processing, error detection and handling, memory utilization and hierarchy, storage management, addressing, control, interrupt handling, and input-output including graphics. Comparison of alternate solutions as implemented in actual systems. Selected new approaches to computer system organization. Prerequisites: Course 170; Course 200 and EE 287 are also desirable.

280. Numerical Analysis of Linear Systems. (3) Computational aspects of linear algebra, linear equations and matrices, direct and indirect methods, eigenvalues and eigenvectors of matrices, error analysis. Prerequisites: Course 180 and Math 190.

281. Numerical Functional Analysis. (3) Foundations of functional analysis: linear spaces, convergence and completeness, operators in Hilbert space. Non-linear operators: iterative methods, fixed point theory, error estimation, monotonicity theory. Applications: boundary value problems, eigenvalue problems, approximation theory, solution of non-linear equations, solution of differential and integral equations. Prerequisites: Math 175 and Math 182 or concurrent enrollment(s).

290. Advanced Topics. (1-3) Arranged as needed to present appropriate material to senior and graduate students. May be repeated for additional credit. Prerequisite: Variable.

294. Human Factors in Computer-Based Systems. (3) Human characteristics relevant for machine design and software design, and for task and job definition; performance control in program and system generation and implementation; social organization characteristics affecting system acceptance and functioning. Prerequisite: Course 170 (Course 200 desirable).

295. Seminar. (1-3) Discussion of and reports on current literature in computer science. May be repeated for additional credit. Prerequisite: Variable.

299. Special Problems. (1-3) Arranged to allow senior and graduate students to pursue work in special problem areas under individual guidance by the staff. May be repeated for additional credit. Prerequisite: Variable.

The Ph.D. Program
in Computer Science
at the
University of Kansas
and
Kansas State University

FOR GRADUATE STUDENTS

310. Theory of Automata. (3) Formal models for computation and their applications, with emphasis on algebraic and logical aspects. Prerequisite: Course 210 or consent of instructor.

314. Theory of Computability. (3) Turing machines, equivalent notions of effective computability, unsolvability results, and computational complexity. Prerequisite: Course 210 or 212, or consent of instructor.

316. Advanced Formal Language Theory. (3) Language manipulation algorithms, recognition, generation and translation strategies. Generalized automata and structure manipulation. Syntax, semantics, interpretation, redundancy, and ambiguity. Complexity of language computation. Other topics as time permits. Prerequisite: Course 216 (Course 314 desirable).

381. Numerical Solution of Nonlinear Operator Equations. (3) Iterative methods for the solution of nonlinear problems: Newton's method and Newton-like methods, generalized secant method, Davidon methods, continuation methods. Convergence analysis: Kantorovich's theorem, error estimation, rates of convergence. Function minimization and nonlinear least squares. Prerequisite: Course 281.

382. Numerical Solution of Ordinary Differential Equations. (3) One step and multistep methods for initial value problems. Stability, consistency and convergence of these methods. Techniques for boundary value problems. Stiff ordinary differential equations and special stability problems. Prerequisite: Math 181.

383. Numerical Solution of Partial Differential Equations. (3) The numerical solution of hyperbolic, parabolic, and elliptic equations by finite difference methods; discretization and round-off errors; the concept of stability for initial value problems; the solution of elliptic boundary value problems by variational and projection methods. Prerequisites: Courses 280 and 281.

385. Numerical Functional Approximation. (3) General finite interpolation, convergence theorems for interpolatory processes, uniform approximation, best approximation, approximation in normed spaces, approximation of linear functionals, spline functions. Prerequisite: Course 281.

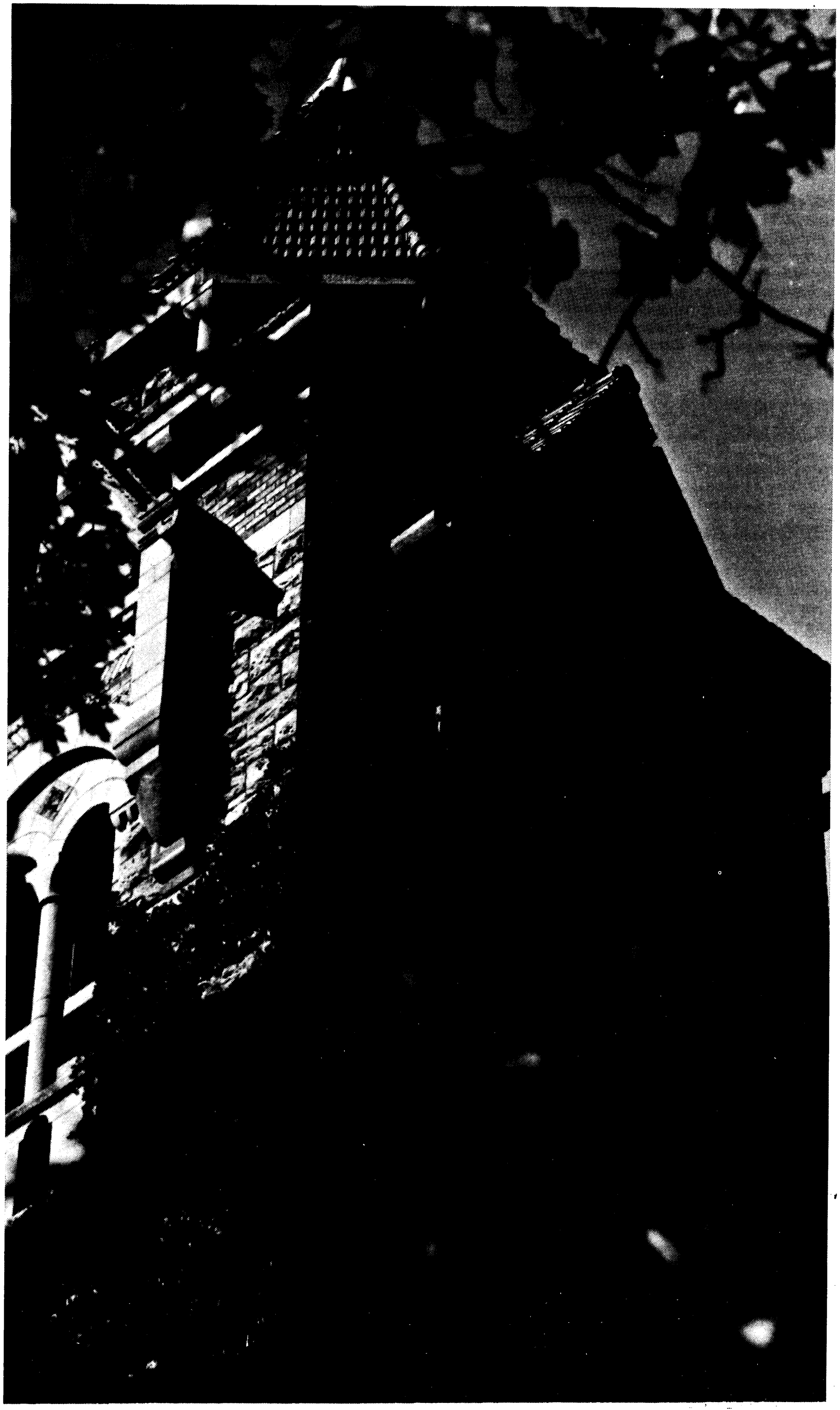
387. Computational Statistics. (3) Computational solution of statistical problems emphasizing distribution functions, special functions, approximations, error analysis, pseudo-random numbers, and verification of random processes. Prerequisites: Courses 180 and 235, and Math 332 or 333.

390. Graduate Topics. (1-3) Arranged as needed to present appropriate material to graduate students. May be repeated for additional credit. Prerequisite: Variable.

395. Research Seminar. (1-3) One or more current research areas in computer science are investigated. Each student will prepare one or more research papers as well as presenting his work orally. May be repeated for additional credit. Prerequisite: Variable.

398. Master's Thesis. (1-6).

399. Doctoral Thesis. (1-10).



Courses of Instruction

AT KANSAS STATE UNIVERSITY

Faculty: Ahmed, Brewer, Calhoun, Conrow, Fisher, Gallagher, Miller, Sackman, Sincovec, Trump, Unger, Weinberg.

FOR UNDERGRADUATE CREDIT ONLY

315. Fundamentals of Computer Programming. (3) Introduction to a procedure-oriented language, the description of a digital computing system, the strategy of problem solving using a digital computer, and the concepts and properties of algorithms. Applications to problem solving. Prerequisite: High School Algebra.

397. Seminar in Computer Science. (arranged)

FOR UNDERGRADUATE AND GRADUATE CREDIT

Note: Courses numbered between 400 and 599 may not be counted toward a graduate degree in Computer Science.

400. Introduction to Algorithmic Processes. (3) Introduction to algorithms, language and notation for describing algorithms, analysis of computational problems and development of algorithms for their solution. The notation of lists, tables, data sets (files) and records. Prerequisite: Course 315.

410. COBOL. (3) Elements of data processing in the COBOL language. Applications. Prerequisite: Course 315.

425. Computer Organization and Programming I. (3) Logical organizations of computers; number systems and arithmetic control units and instruction sequencing, assemblers, addressing systems, subroutine linkages (transfer vectors), and input-output operations. Prerequisite: Course 400, or consent of instructor.

440. Introduction to Programming Languages. (3) Structure of algorithmic languages. Conversational languages. List processing and string manipulation languages. Concepts and facilities of programming languages. Prerequisite: Course 400.

505. Mathematical Machines and Computability I. (4) Elements of matrix algebra pertinent to digital computations. Computer methods of solving linear equations and inverting matrices. Error analysis, problem conditioning and post-optimizations. Rectangular and singular systems. A generalized inverse for matrices. Algorithmic methods of solving eigenvalue problems, progressive algorithms. Applications. Three hours lecture, one hour laboratory on a digital computer. Prerequisite: Math. 221 and Course 315.

506. Mathematical Machines and Computability II. (4) Computer algorithms for finding roots of polynomials and the real roots of transcendental equations. Error analysis, effect of uncertainty in the coefficients. Computer algorithms for the approximation of continuous functions. Numerical integration, differential equations. Three hours lecture and one hour laboratory on a digital computer. Prerequisite: Course 505 or Math. 223 and Course 315.

525. Computer Organization and Programming II. (3) Study of information representation and processing techniques. Transformations between storage media. Referencing of information as related to the structure of its representation and implications for the design of the referencing language. Prerequisite: Course 425.

535. Non-numeric Programming. (3) The use of computers in areas not involving numeric calculations. Surveys of applications into areas such as music, learning theory, games and discrete pattern recognition. Heuristic programs. Prerequisite: Course 400.

597. Seminar in Computer Science. (arranged)

600. Discrete Structures. (3) Study of linear lists, strings, arrays, orthogonal lists, and graphs. Representation of such structures within computers. Prerequisite: Course 400.

610. List and String Processing Languages. (3) The languages LISP and SNOBOL will be studied, and problems will be developed illustrating the use of each. Prerequisite: Courses 400 and 600.

615. Theory of Computability. (3) Propositional calculus, axiomatics. Turing machines; unsolvable problems; quantification theory; satisfiability and validity models, first-order theories; foundational considerations. Prerequisite: E.E. 355, or consent of instructor.

620. Programming Systems. (3) Languages for writing software, design of assembly systems, macro-instructions, operating systems (monitors), interrupt systems, storage allocation, and multiprogramming. Prerequisite: Courses 400 and 600.

631. Numerical Solution of Ordinary Differential Equations. (2) (Concurrent with Math 631) Computer algorithms and techniques for solving ordinary differential equations. Programming exercises on the digital computer. Prerequisite: Math 555, Computer Science 315 or Computer Science 505, Math 240 plus concurrent enrollment in Math 631.

632. Numerical Solution of Partial Differential Equations. (2) (Concurrent with Math 632) Computer algorithms and techniques for solving partial differential equations. Programming exercises on the digital computer. Prerequisite: Math 631, Computer Science 631 plus concurrent enrollment in Math 632.

635. Artificial Intelligence. (3) Application of heuristics to problem solving. Perceptrons, pattern recognition, learning, self-evolving programs. Prerequisite: Course 535.

640. Programming Languages. (3) The study of the structure and facilities of major algorithmic procedure-oriented languages and their implementation, operation and use. Prerequisite: Courses 440 and 600.

670. Information Organization and Retrieval. (3) Models for representing structured information, techniques for organizing and searching files. Structure of semiformal languages. Analysis of information by statistical, syntactic and logical methods. Applications to automatic information retrieval systems, question answering systems, and man-machine interaction. Prerequisite: Course 425.

701. Automata Theory. (3) Finite automata; synchronous sequential circuits; Kleene's Theorem; semi-groups; monomorphisms; generator systems; algebraic linguistics; potentially infinite machines; theory of computability, recursive functions; programming systems. Prerequisite: Course 615 and Math. 512, or consent of instructor.

710. Compiler Design I. (3) Formulation of syntax-directed and table-driven techniques used in compiler design. Various alternative techniques. Environment of a compiler, conversational compilers. Prerequisite: Courses 620 and 640.

711. Compiler Design II. (3) Conversational compilers. Syntax-directed compilers. Extensible compilers. Compiler writing systems. Prerequisite: Course 710.

712. Seminar in Computer Science. (1)

720. Business Data Processing. (4) Manual, semi-automatic, automatic systems of data processing. Accounting concepts, data processing implications. Organization of sequencing and direct-access files. Checking and control techniques. Student groups will study business applications and recommend data-processing systems. Three hours lecture, two hours lab each week. Prerequisite: Courses 410 and 600.

760. Computers, Science and Society. (3) Critical review of computers and social problems. Impact of computers on science and experimental method. Guided research on some aspect of the social use of computers selected by each student. Prerequisite: Consent of the instructor.

797. Seminar in Computer Science. (arranged)

FOR GRADUATE STUDENTS

800. Computational Semantics. (3) Theoretical prerequisites and computational techniques for mechanical interpretation of language sentences. Semantics of formal computer languages, including query languages for information retrieval. Structural representation of meaning. Prerequisite: Course 615.

810. Computer Simulation. (3) A variety of examples will be studied to illustrate the power and flexibility of automata, theoretic, representation, statistical techniques, and information theory in simulation studies. Prerequisite: Course 620.

811. Computer Simulation Experiments. (3) A computer simulation will be programmed for the digital computer and used to predict data and test hypotheses. Prerequisite: Course 810, or consent of instructor.

815. Special Topics in Computer Science. (2-4) Study in selected areas of artificial intelligence, computational linguistics, linear and nonlinear programming, theorem proving by computer, models of intelligent processes, and the like. Prerequisite: Consent of instructor.

865. Computer Simulation of Eco-systems. (3) A selection of various eco-systems will be used to show how a computer can be used to formally define an eco-system. Then computer simulated eco-systems will be used to discover ways to optimize the benefits to be derived from actual eco-systems. Prerequisite: Course 810, or consent of instructor.

998. Research in Computer Science and Mass Communication. (arranged)

Advanced Research

IV. ADVANCED RESEARCH

Both Kansas State University and the University of Kansas have recruited strong faculties, and the student can avail himself of one or both as he feels the need or inclination. Although each faculty covers the basic areas of computer science, there are distinctions between the programs at the Ph.D. research level. Currently, the identification of specialties is as follows:

Identified with the University of Kansas

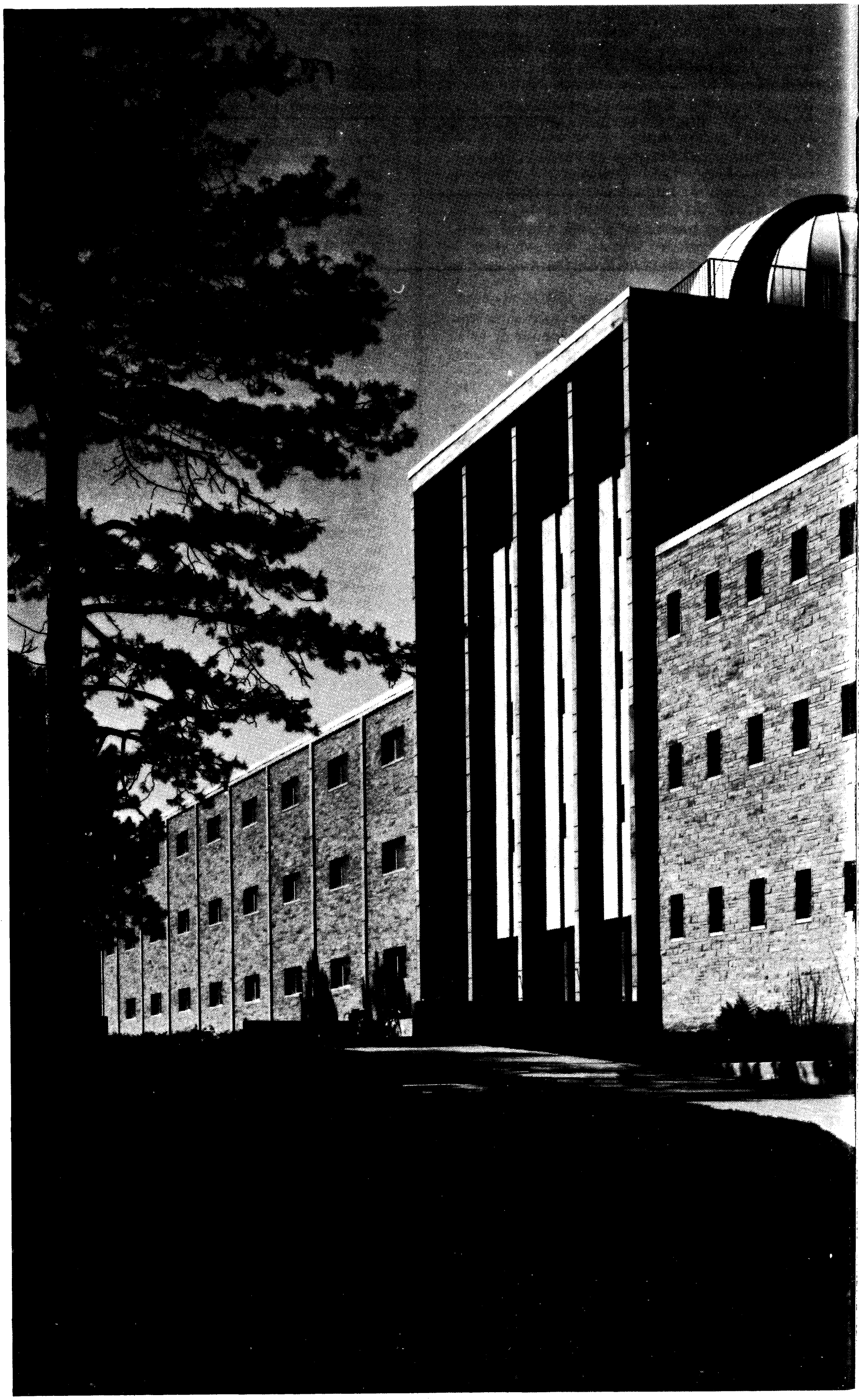
- Formal Language Theory—Theory of grammars, formal languages, formal semantics.
- Natural Languages and Symbol systems—Computational linguistics, pattern generation in the humanities and fine arts, sound synthesis and analysis.
- Automata and Mathematical Logic — Theory of automata, computability, recursive function theory.
- Machine Systems.
- Information Systems Theory and Design—Analysis of information networks, information acquisition, social implications of information systems.

Areas of Current and Essential Interest to Both Campuses

- Numerical Analysis.
- Artificial Intelligence.

Identified with Kansas State University

- Machine Languages—Language processors, conversational languages, extensible languages, SEMANTICS
- Computer Design and Architecture~~—Computer logic, switching theory.~~
- Programming Systems.
- ~~Biological and Ecological~~ Systems Simulation, SYSTEMS
- Data Organization and Manipulation—File management and data processing, information storage and retrieval, text processing.



Faculties and Specialties

COMBINED FACULTIES

Nasir Ahmed, Associate Professor (KSU) (jointly with Electrical Engineering) *
Ph.D., Electrical Engineering, New Mexico, 1966
Scientific data processing techniques, pattern classification, communication theory, information theory, network analysis

Zamir Bavel, Associate Professor (KU) *
Ph.D., Mathematics, Illinois, 1965
Automata theory

Paul T. Boggs, Assistant Professor (KU) *
Ph.D., Computer Science, Cornell, 1970
Numerical analysis

Richard Brewer, Assistant Professor (KSU)
M.S., Journalism, University of Wisconsin, 1964
Computational linguistics and information retrieval

William G. Bulgren, Associate Professor (KU) (jointly with Computation Center) *
Ph.D., Statistics, Iowa, 1965
Languages, simulation, computational statistics

Myron A. Calhoun, Assistant Professor (KSU) (jointly with Electrical Engineering) *
Ph.D., Electrical Engineering, Arizona State, 1967
Hardware design and implementation systems

John Case, Assistant Professor (KU) *
Ph.D., Mathematics, Illinois, 1969
Recursive functions, logic, automata

Kenneth Conrow, Associate Professor (KSU) * ✓
Ph.D., Organic Chemistry, Illinois, 1957
Computer generation of organic nomenclature, computer applications to lattice theory

Paul S. Fisher, Assistant Professor (KSU) *
Ph.D., Computer Science, Arizona State University, 1969
Programming languages, artificial intelligence and systems

Tom L. Gallagher, Associate Professor (KSU) (Director, Computing Center) *
D.Sc., Applied Mathematics and Computer Science, Washington University, 1967
Mathematical programming, information storage and retrieval and biomedical data processing

Richard G. Hetherington, Associate Professor (KU) (jointly with Mathematics and Computation Center) *
Ph.D., Mathematics, Wisconsin, 1961
Numerical analysis

Floyd R. Horowitz, Associate Professor and Acting Chairman (KU) *
Ph.D., English, Iowa, 1960
Language analysis and humanities computation

Lois E. Mansfield, Assistant Professor (KU) *
Ph.D., Mathematics, Utah, 1969
Numerical analysis

Michael H. Miller, Assistant Professor (KSU) (assistant director, Computing Center)
M.S., Statistics, Iowa State, 1961
Operating systems, programming languages

Harold Sackman, Professor and Head (KSU) *
Ph.D., Psychology, Fordham, 1953
Man-computer problem solving, real time systems, online planning, mass information utilities, computer-aided social experimentation

Earl J. Schweppe, Professor (KU) *
Ph.D., Mathematics, Illinois, 1955
Systems, languages, data structures

Sally Yeates Sedelow, Professor (KU) (jointly with Linguistics) *
Ph.D., English, Bryn Mawr, 1960
Humanistic and linguistic computation, pattern generation and recognition

Walter A. Sedelow, Jr., Professor (KU) (jointly with Sociology) *
Ph.D., History, Harvard, 1957
Language analysis, human factors, information systems, and public applications

Richard F. Sincovec, Assistant Professor (KSU) *
Ph.D., Applied Mathematics, Iowa State University, 1968
Numerical analysis

Thomas N. Trump, Assistant Professor (KSU)
M.S., Computer Science, Purdue University, 1966
Systems programming and numerical analysis

Douglas Tuggle, Assistant Professor (KU) (jointly with Business School) *
Ph.D., Industrial Administration, Carnegie Mellon, 1970
Cognitive processes

Elizabeth A. Unger, Assistant Professor (KSU) (Associate Director, Computing Center)
M.S., Mathematics, Michigan State University, 1963
Information storage and retrieval programming language, program libraries

Roger Weinberg, Associate Professor (KSU) *
Ph.D., Zoology (Genetics), University of Texas, 1954
Ph.D., Computer Science, University of Michigan, 1970
Computer simulation of eco-systems and computer systems, model simplification for purposes of simulation

* Authorized to direct doctoral dissertation research.

The Ph.D. Program in Computer Science at Kansas State University and the University of Kansas

Facilities and Resources

V. FACILITIES AND RESOURCES

Libraries:

Both Kansas State University and the University of Kansas have been expanding their holdings in computer science and closely related fields at a rate commensurate with the rapid expansion of this active discipline. Thus the doctoral student will have access to excellent library resources in all areas of computer science. Accessibility is enhanced by a courier service which operates between the libraries.

Computers:

At the University of Kansas

The main general purpose computer at the University of Kansas is a GE/Honeywell 635 with 200,000 36-bit words of memory, a complete collection of peripherals including eight removable disc storage units and eight magnetic tape drives providing the user the ability to have tapes and discs for his personal work. The 635 is interfaced through a communications processor, the Datanet 30. There are 50 remote interactive terminals on campus which include Teletype mod 33, Terminet 300, Datanet 760, Datapoint 3300, Datapoint 3000, and Datapoint 2200 terminals. A variety of these terminals is available in a single classroom for teaching and research activities. An IBM 1401 with 8,000 characters of core memory and four tape drives, a stand-alone digital incremental plotter with a 36-inch drum and the usual complement of unit record equipment are also available to support research and teaching activities. Four special purpose computers are installed on campus and are available by arrangement with the local departments. These include an IBM 1800 in the Physics Department, two Hewlitt-Packard 2116Bs in the Chemistry Department, and a PDP 15/20 in the Electrical Engineering Department.

The 635 provides local batch processing, remote batch processing and time-sharing concurrently using an integrated file system. In addition to powerful systems software, 15 language processors and more than 20 applications program packages are available to the user. Detailed descriptions of services and procedures are contained in the booklet "Guide for Using Computer Facilities," available from the Kansas University Computation Center.

At Kansas State University

The main general purpose computer at Kansas State University is an IBM System 360/50 with 131,000 bytes of fast core and 1 million bytes of slow core. It also has removable disc drives, tape units, and a four line communications adapter. There are six interactive terminals including IBM 2741's and a Datel 30. An IBM 1230 Optical Mark Sense Reader and Calcomp 663 Digital Incremental Plotter are available at the Kansas State University Computing Center. Four additional digital computers are installed on campus and are available by arrangement with the local department. These include a Univac Athena and a NOVA computer in the Department of Electrical Engineering, a PDP 15/30 in the Department of Physics, and a NOVA computer in the Department of Computer Science.

The IBM S360/50 has the purpose of providing the academic community with instructional and instructional-support computing service. The machine is partitioned to provide batch processing, self-service batch processing, and time-sharing support for the communications terminals. It is run using IBM's Operating System with the Houston Automatic Spooling Priority (HASP) program to provide accounting and service the job queue, and the Baylor Executive System for Teleprocessing (BEST) to provide the time-sharing environment for the terminals. Seventeen language processors and 15 major applications packages are presently available to the user. Detailed descriptions of services and procedures are contained in the "Kansas State University Computing Center Users Guide," which is available from the Computing Center.

DECLARATION OF INTENT
to enter the
COMPUTER SCIENCE DOCTORAL PROGRAM
Jointly Offered by the
UNIVERSITY OF KANSAS AND KANSAS STATE UNIVERSITY

(Complete and send to Chairman, Graduate Studies Committee, at Institution you name below)

Name ____ Date ____________________

Social Security Number ____

Address ____

I hereby apply for acceptance as an aspirant in the Doctor of Philosophy Degree Program in Computer Science jointly offered by the Computer Science Departments at the University of Kansas and Kansas State University. I presently plan to complete my work for this degree at

____
(Name of Institution)

Remarks:

Date ___________________________ Signature ______________________________________
(Applicant)

The above graduate student in Computer Science is hereby recommended as an aspirant for the Doctor of Philosophy Degree in Computer Science.

Remarks:

Date __________________________ Authorized Signature _________________________________
(Chairman, Graduate Studies, Named Institution)

The above graduate student in Computer Science is hereby accepted as an aspirant for the Doctor of Philosophy Degree in Computer Science.

Remarks:

Date __________________________ Authorized Signature _________________________________
(Chairman, Graduate Studies, Other Institution)