The Interdisciplinary Ph.D. Program
in Pure and Applied Logic
at
Carnegie Mellon University
Fall 2002

1 Nature and aims of the program

Carnegie Mellon’s Ph.D. Program in Pure and Applied Logic is an interdisciplinary venture, jointly
sponsored by the Departments of Mathematical Sciences, Philosophy, and Computer Science. Each
of these three departments administers a track of the Program.

Carnegie Mellon has a large and active group of faculty in Pure and Applied Logic, with a par-
ticularly strong concentration in foundational aspects of Computing. This Logic Community has
an established record of collaborations in pursuing theoretical research, conducting major imple-
mentation projects, and running colloquia and workshops. The Program builds on these strengths
to educate new generations of scientists who will pursue research in Pure and/or Applied Logic.

The PAL World-Wide Web home page is at
http://www.cs.cmu.edu/~pal
2 The faculty

As of Fall 2002, the Program faculty consists of:

- **Clark Glymour**, Alumni University Professor of Philosophy;
- **Dana Scott**, Hillman University Professor of Computer Science, Philosophy, and Mathematical Logic;
- **Teddy Seidenfeld**, Herbert A. Simon Professor of Philosophy and Statistics;
- **Peter Andrews**, Professor of Mathematics;
- **Henk Barendregt**, Adjunct Professor, Mathematical Sciences;
- **Cristina Bicchieri**, Professor of Philosophy and Social and Decision Sciences;
- **Lenore Blum**, Distinguished Career Professor of Computer Science;
- **Edmund Clarke**, FORE Systems Professor of Computer Science;
- **Robert Harper**, Professor of Computer Science;
- **Kevin Kelly**, Professor of Philosophy;
- **Peter Lee**, Associate Professor of Computer Science;
- **John Reynolds**, Professor of Computer Science;
- **Wilfried Sieg**, Professor of Philosophy, Head of the Philosophy Department;
- **Peter Spirtes**, Professor of Philosophy;
- **Richard Statman**, Professor of Mathematics and Computer Science;
- **Stephen Brookes**, Associate Professor of Computer Science;
- **James Cummings**, Associate Professor of Mathematics;
- **Rami Grossberg**, Associate Professor of Mathematics;
- **Michael Kohlhase**, Adjunct Associate Professor, Computer Science and Language Technologies;
- **Frank Pfenning**, Associate Professor of Computer Science;
- **Richard Scheines**, Associate Professor of Philosophy;
- **Horacio Arlo-Costa**, Assistant Professor of Philosophy;
- **Jeremy Avigad**, Assistant Professor of Philosophy;
- **Steve Awodey**, Assistant Professor of Philosophy;
- **Peter Vanderschraaf**, Assistant Professor of Philosophy;
- **Ernest Schimmerling**, Assistant Professor of Mathematics;
- **Mandy Simons**, Assistant Professor of Philosophy and Linguistics;
Logicians at the adjacent University of Pittsburgh include N. Belnap, R. Daley, K. Manders, and R. Thomason.

Current research interests of the Program’s faculty include: Automated Theorem Proving, Category Theory, Constructive and Feasible Mathematics, Decision Theory, Foundations of Programming Languages, Game Theory, Logics of Programs, Lambda Calculus, Learning Theory, Model Theory, Proof Theory, Set Theory, Set-theoretic Algebra, Temporal and Modal Logics, Theory of Computing, and Type Theory.

The Program is further strengthened by major research and educational activities at Carnegie Mellon in related fields, such as Algorithms, Artificial Intelligence, Combinatorial Optimization, Computational Complexity, Computational Linguistics, Operations Research, and Programming Systems.

Here is a brief description of the research interests of the Program faculty.

- **Clark Glymour, Alumni University Professor of Philosophy**
  Professor Glymour has worked in several areas of philosophy of science, including the philosophy of physics, theories of scientific explanation and confirmation, philosophy of psychology, and topics in philosophical logic. His present research interests include computational learning theory, methodological issues in artificial intelligence, formal methods in automated causal inference, and methodological issues in cognitive psychology. He is now at work on a book on Freud and Cognitive Science.

- **Dana Scott, Hillman University Professor of Computer Science, Philosophy, and Mathematical Logic**
  Professor Scott’s work in logic has concerned the theories of models, automata and sets; modal and intuitionistic logic, constructive mathematics, and connections between category theory and logic. His interests in Philosophy concern the foundation and philosophy of logic and mathematics and the semantical analysis of natural language. His work in computer science has been principally directed toward the development of denotational semantics of programming languages and mathematical foundations for computability. Currently he aims at unifying the semantical approach with constructive logical formalisms to be able to give rigorous and machine implementable proof methods and development tools for the construction of correct programs. He also has a strong interest in promoting the use of computer-algebra systems in teaching and research.

- **Teddy Seidenfeld, Herbert A. Simon Professor of Philosophy and Statistics**
  Professor Seidenfeld is interested in statistical inference involving the use of merely finitely additive probabilities, and their surprising effect on statistical decision theory. He is also interested in the general question of whether the norms of Bayesian statistics can be extended from individuals acting alone to cooperative groups. His goal is to provide a unified treatment that encompasses both individual and cooperative group decision making. The strategy that he employs is to relax a central assumption of Bayesian expected utility theory, the ”ordering” postulate. This is joint research with colleagues of his from the Statistics department.

- **Peter Andrews, Professor of Mathematics**
  Professor Andrews’ interests center on automated theorem proving, especially for theorems of type theory (higher-order logic). Research is based on an approach to automated theorem proving involving expansion trees and matings, which represent the essential combinatorial structure inherent in proofs of theorems. They can be found by heuristic search, and natural deduction proofs can be constructed from them without further search. A computer system called TPS implementing these ideas in Common Lisp has been developed. It can be used in automatic or interactive mode, or a combination of these, to construct proofs and to translate proofs from one format to another. A fragment of TPS called ETPS is used as an interactive aid by students in logic courses.
• Henk Barendregt, Adjunct Professor, Department of Mathematical Sciences  
  Lambda calculus, Logic, Type theory, Formalizing mathematics  
  Lambda calculus is a language capable of expressing algorithms with clear operational and denotational semantics. Also it forms a compact language to denote mathematical proofs. Logic provides a formal language in which mathematical statements can be formulated and provides deductive power to derive these. Type theory is a formal system, based on lambda calculus and logic, in which statements, computable functions and proofs all can be naturally represented. As such it is a good medium to represent mathematics on a computer with the aim to exchange and store reliable mathematical knowledge. The preferred levels of deductive and computational power can be set as parameters in a natural way.

• Cristina Bicchieri, Professor of Philosophy and Social and Decision Sciences  
  Professor Bicchieri’s research centers around two main topics: game theory and philosophy of social science. Her work on game theory includes the logical foundations of equilibrium, belief revision models, applications of epistemic logic and nonmonotonic logics to game modeling and applications of game theory to distributed AI. Her work in the philosophy of social science is primarily on the emergence and evolutionary dynamics of social norms.

• Lenore Blum, Distinguished Career Professor of Computer Science  
  Professor Blum’s early work in logic was in the interface between model theory and algebra. Her more recent work in the theory of computation lies in the interface between the continuous and the discrete. In 1989, along with Steve Smale and Mike Shub, she introduced a theory of computation and complexity over an arbitrary ring or field R. Their viewpoint is: While the classical theory of computation has been extraordinarily successful in providing the foundations and framework for theoretical computer science, it’s dependence on 0’s and 1’s, is fundamentally inadequate for providing such a foundation for modern scientific computation, where most of the algorithms - with origins in Newton, Euler, Gauss, et al. - are real number algorithms. Blum’s current research interests focuses on interconnections between model theory and computational complexity theory, particularly on transfer principles for complexity problems such as “does P = NP?”

• Edmund Clarke, FORE Systems Professor of Computer Science  
  Professor Clarke works on the boundary between theoretical computer science and programming systems. He is particularly interested in the use of formal methods for constructing correct and reliable hardware and software. His research group has developed a temporal logic based verifier for finite state systems that can automatically check properties of sequential circuits with more than $10^{20}$ reachable states and has been used to find subtle errors in hardware controllers. He is also interested in developing powerful theorem provers that are able to handle theorems with significant complexity.

• Robert Harper, Professor of Computer Science  
  Professor Harper’s research activities center around two main topics: programming languages and computer-assisted formal reasoning. His work in programming languages focuses primarily on the role of type structure in language design, and on the use of operational semantics as the basis for the definition and implementation of programming languages. His work in computer-assisted formal reasoning is concerned with the development of the theory and practice of general logical frameworks as the basis for implementing formal systems on a computer.

• Kevin Kelly, Professor of Philosophy  
  Professor Kelly’s interests lie primarily in the area of epistemology and the philosophy of science. His current research interests focus on computational learning theory as a response to the problem of induction and as a tool to discern the scientific limitations of computation.
• **Peter Lee, Professor of Computer Science**
  Professor Lee’s research interests lie in the area of semantics-based analysis and implementation of programming languages. He is particularly interested in the use of logic and semantics to specify optimizing compilers.

• **John Reynolds, Professor of Computer Science**
  Professor Reynolds is interested in the design of programming languages and languages for program specification, mathematical tools for defining the semantics of such languages, and methods for proving that programs satisfy their specifications. His current goals are (1) to understand and unify the rich variety of type systems, including polymorphic types, intersection types, and the types of linear logic, that have proliferated in recent years and (2) to extend strong typing and proof systems to low-level languages that give the user control over data representation and storage allocation.

• **Wilfried Sieg, Professor of Philosophy, Head of the Philosophy Department**
  Professor Sieg’s main research area is in Mathematical Logic and Philosophy of Mathematics. He has done mathematical work in proof theory, and has also given philosophical analyses of foundational approaches in the 19th and 20th century. His interests include closely related issues in the foundations of Cognitive Science (Church-Turing thesis), applications of logic to automated proof search, and the educational use of sophisticated (machine) tutors.

• **Peter Spirtes, Professor of Philosophy**
  Professor Spirtes (in collaboration with Clark Glymour and Richard Scheines) has been investigating how to reliably infer causal relations from statistical data, given various kinds of background information. By connecting causal hypotheses with statistical hypotheses of conditional independence and graphical representations of statistical models, we can precisely delineate what background knowledge is required in order to reliably infer various features of the causal relations between random variables. We have developed, proved the correctness of, and implemented a number of algorithms for reliable causal inference that can be run on as many as a hundred variables, as long as the causal relations between the variables are sparse. We are currently turning these algorithms into a practical tool (TETRAD II) that will be useful to statisticians and in the development of expert systems.

• **Richard Statman, Professor of Mathematics and Computer Science**
  Professor Statman’s interests include mathematical logic (λ-calculus, proof theory, constructivity, theory of computation), combinatorics and graph theory (especially topological graph theory, well partial orderings, coin weighing problems and combinatorial algorithms), as well as theoretical computer science (complexity theory, denotational semantics, theorem proving, foundations, logic of programs).

• **Stephen Brookes, Associate Professor of Computer Science**
  Professor Brookes works generally on issues related to the semantics of programming languages, including the development of new semantic models and semantically based techniques for reasoning about program behavior. His current focus is on intensional semantics: models of programs which permit reasoning about efficiency as well as correctness. He is particularly interested in developing compositional proof methods for reasoning about parallel programs.

• **James Cummings, Associate Professor of Mathematics**
  Professor Cummings’ research interests are centered in set theory, especially the areas of large cardinals, forcing and inner models. His recent work includes results on stationary reflection, combinatorial principles at successors of singulars, and \( \aleph_2 \)-Souslin trees. He is also interested in dynamical systems, and in the application of logic to computer science.
• Rami Grossberg, Associate Professor of Mathematics
  Professor Grossberg’s research interests are concentrated in pure and applied model theory, especially in classification theory of the model theory of infinitary logics, related problems in combinatorial set theory, and applications of these to algebra.

• Michael Kohlhase, Adjunct Associate Professor of Computer Science and Language Technologies
  Dr. Kohlhase’s research interests lie in higher-order unification and theorem proving, logics and calculi for partial functions, deduction systems as a whole, and finally natural language semantics. He has been centrally involved in the developed of Omega, a mathematical proof assistant, MBase, a distributed mathematical knowledge base, and OMDoc, an emerging XML-based markup standard for structured mathematical documents.

• Frank Pfenning, Associate Professor of Computer Science
  Professor Pfenning’s research interests include logic programming and functional programming and the application of methods from mathematical logic and type theory to the problem of programming language description and implementation. In particular, he has been working on a logical framework, that is, a language for the definition and manipulation of formal systems such as various logics, type systems or operational semantics. He is also interested in theorem proving in higher-order logics.

• Richard Scheines, Associate Professor of Philosophy
  Dr. Scheines' interests are in the relationship between causal knowledge and statistical data. Along with Peter Spirtes and Clark Glymour, Prof. Scheines has studied algorithms for inferring features of causal systems from statistical data. He is also interested in educational applications of this theory, in particular computerized education. Having previously developed the Carnegie Mellon Proof Tutor, he has recently built a web-based course on causal reasoning with statistical data.

• Horacio Arlo-Costa, Assistant Professor of Philosophy
  Professor Arlo-Costa’s interests have focused on epistemology and knowledge representation (in AI and economic modelling). This work has been motivated by an interest in understanding and clarifying the nature of rationality. Recent research centered around methodological problems in AI and conditional logic. Current research includes work in the foundations of cognitive science, causal modelling, and models of bounded rational agents.

• Jeremy Avigad, Assistant Professor of Philosophy
  Professor Avigad studies mathematical logic. His particular research interests include proof theory, constructive mathematics, proof complexity, and the history and philosophy of mathematics. He is especially interested in understanding the relationship between classical and constructive aspects of mathematical practice.

• Steve Awodey, Assistant Professor of Philosophy
  Professor Awodey’s interests are in logic, its relationship to mathematics and computer science, and the surrounding philosophical issues. The focus of his current research is topos theory and higher-order logic. His technical work is mainly in the areas of type theory, categorical logic, and sheaf theory. Category theory is his chief technical tool; it also informs his perspective on philosophical issues in mathematics.

• Peter Vanderschraaf, Assistant Professor of Philosophy
  Professor Vanderschraaf is working on the development of a general theory of convention, and proposes to use this theory to address longstanding difficulties in ethics and political philosophy. As part of his overall project, he is attempting to fully integrate the insights into the nature of convention of both the philosophical and the social science traditions.
• Ernest Schimmerling, Assistant Professor of Mathematics
  Professor Schimmerling’s work has been in set theory, with an emphasis on core models and their connections to other areas such as descriptive set theory and infinitary combinatorics.

• Mandy Simons, Assistant Professor of Philosophy and Linguistics
  Professor Simons’ research deals with the formal semantics and pragmatics of natural language. She is particularly interested in phenomena which show the interaction of compositional semantics with general conversational principles. To date, her work has examined linguistic expressions whose interpretation requires a cross-sentential perspective, or reference to context: pronouns, definite descriptions, and presupposition-inducing expressions. Currently, she is working on the semantics and pragmatics of natural language disjunction.
3 Courses, colloquia, and conferences

The following graduate courses and seminars were offered by the Carnegie Mellon Logic faculty in the recent past or are planned for the next academic year. Courses with prefix 15 are offered by the Computer Science Department, those with prefix 21 are offered by the Department of Mathematical Sciences, and those with prefix 80 are offered by the Philosophy Department.

- **21-600 Mathematical Logic I**
  The study of formal logical systems which model the reasoning of mathematics, scientific disciplines, and everyday discourse. Propositional calculus and first-order logic. Syntax, axiomatic treatment, derived rules of inference, proof techniques, computer-assisted formal proofs, normal forms, consistency, independence, semantics, soundness, completeness, the Löwenheim-Skolem Theorem, compactness, equality. Offered every fall.

- **80-600 Minds, Machines, & Knowledge**
  We examine important presuppositions in several contemporary essays that debate the scope and limits of artificial intelligence. Specifically, the class discusses contemporary views on foundational versus coherence models of human knowledge. (Does human knowledge have a secure basis or is it a mere "web of beliefs"?) These questions about the organization of knowledge are contrasted with objections raised, for example, by Searle and Dreyfus against the "strong AI" thesis of Newell and Simon. Also, we consider several current models of knowledge which incorporate probability and other measures of uncertainty, including some recent work on "parallel" systems.
  Prerequisite: 80-100 or some equivalent introduction to Philosophy

- **21-602 Introduction to Set Theory**
  This is a standard first graduate course in set theory. The two main topics are constructibility and forcing. We will cover chapters I and III-VII of the textbook in order, and refer back to chapter II as needed. Our primary goal is the independence of the continuum hypothesis.
  Textbook. Kenneth Kunen *Set Theory: An Introduction to Independence Proofs*
  Prerequisites. Students should have a background in undergraduate level set theory (cf. 21-229 and the textbook by Hrbacek & Jech) and logic (cf. what I covered in 21-700 or the logic textbooks by Goldstern & Judah or Enderton), which includes a working knowledge of basic ordinal and cardinal arithmetic, Gödel’s completeness theorem, and the downward Löwenheim-Skolem theorem. An understanding of the statement of Gödel’s theorem on consistency proofs will also be assumed.

- **21-603 Model Theory I**
  Contents include: Similarity types, structures. Downward Löwenheim-Skolem theorem. Construction of models from constants, Henkin’s omitting types theorem, prime models. Elementary chains of models, some basic two-cardinal theorems, saturated models, basic results on countable models including Ryll-Nardzewski’s theorem. Indiscernible sequences, Ehrenfeucht-Mostowski models. Introduction to stability, various rank functions, primary models, and a proof of Morley’s categoricity theorem. Basic facts about infinitary languages, computation of Hanf-Morley numbers.
  Prerequisite: Elementary logic or permission of the instructor
  Text: Lecture notes by the instructor

- **80-605 Rational Choice**
  This course will cover selected topics in rational choice theory, which informally is the analysis of how to make a correct decision in a given context. Possible topics may include, and are not limited to: individual decision making under uncertainty, problems of public choice in which a group of individuals must collectively make a decision, game-theoretic problems of conflict and coordination, and alternative approaches to the problem of fair division of goods. This course will stress the role that formal methods can play in the analysis of decisions and alternative applications of decision theory to issues in philosophy and social science.
• **80-610 Logic and Computation**

Among the most significant developments in logic in the twentieth century is the formal analysis of the notions of provability and semantic consequence. For first-order logic, the two are related by the soundness and completeness theorems: a sentence is provable if and only if it is true in every interpretation. This course begins with a formal description of first-order logic, and proofs of the soundness and completeness theorems. Other topics may include: compactness, the Lwenheim-Skolem theorems, nonstandard models of arithmetic, definability, other logics, and automated deduction. Prerequisites: 80-210 or 80-211, or an equivalent course in Mathematics or Computer Science.

Prerequisites: either 80-210, 80-211, or an equivalent course in Mathematics or Computer Science.

• **80-611 Computability & Incompleteness**

The 1930’s witnessed two revolutionary developments in mathematical logic: first, Gdel’s famous incompleteness theorems, which demonstrate the limitations of formal mathematical reasoning, and second, the formal analysis of the notion of computation in the work of Turing, Gdel, Herbrand, Church, Post, Kleene, and others, together with Turing’s results on the limits of computation. This course will cover these developments, and related results in logic and the theory of computability. Prerequisites: 80-210 or 80-211, or an equivalent course in Mathematics or Computer Science.

• **80-612 Philosophy of Mathematics**

Problems in the foundations of mathematical analysis prompted, around the turn of the century, renewed philosophical reflection on mathematics and related mathematical and logical work. Set theoretic and constructivist foundations for analysis are described; Hilberts program is analyzed in detail. Finally, a structuralist position is defended: this position is philosophically influenced by the work of Bernays and mathematically by developments in mathematics (as reflected in Bourbaki’s treatises and category theory); it incorporates important insights from the main foundational schools.

• **80-614 Logic in Artificial Intelligence**

An introduction to several formalisms used in knowledge representation and database theory. The emphasis is placed on nonmonotonic logic, conditional logic and belief revision methods. We will also study recent issues in the logics of knowledge and belief and consider applications in distributed AI. Several methodological problems in AI are discussed. Prerequisites: A basic course in logic is recommended but not required.

• **80-615 Modal Logic**

An introduction to first-order modal logic. The course considers several modalities aside from the so-called alethic ones (necessity, possibility). Epistemic, temporal or deontic modalities are studied, as well as computationally motivated modals (like ’after the computation terminates’). Several conceptual problems in formal ontology that motivated the field are reviewed, as well as more recent applications in computer science and linguistics. Kripke models are used throughout the course, but we also study recent Kripkean-style systematizations of the modals without using possible worlds. Special attention is devoted to Scott-Montague models of the so-called ‘classical’ modalities. Prerequisites: 80-210, or 80-211, or instructor’s permission.

• **80-616 Probability & Artificial Intelligence**

In this course we will examine foundational questions about the concepts of causality and probability, how artificial intelligence techniques can be used to solve some of the computational problems presented by the use of probabilities and representations of causal relations, and how probabilities and representations of causal relations have been incorporated into recently developed expert systems. The foundational questions we will examine are: What do causal and probabilistic statements mean? How can probabilities and causal relations be inferred? Are there any axioms relating causal relations to probability distributions? What are the advantages and disadvantages of using probabilities as compared to alternative representations of uncertainty? We will then discuss recent developments in Artificial Intelligence (e.g. Bayesian networks) which have solved some of the long-standing computational problems associated with the use of probabilities and statements about causal relations. Finally, we will study in detail some expert systems, such as QMR and Pathfinder, which have incorporated these new techniques in order to perform medical diagnosis.
• **80-617 Constructive Logic**

This multidisciplinary junior-level course is designed to provide a thorough introduction to modern constructive logic, its roots in philosophy, its numerous applications in computer science, and its mathematical properties. Some of the topics to be covered are intuitionistic logic, inductive definitions, functional programming, type theory, realizability, connections between classical and constructive logic, decidable classes.

Prerequisites: 80-210 or 80-211 and a programming course

• **80-620 Philosophy of Physics (Topics in the Philosophy of Science)**

The course will not presuppose any particular course work in physics, but will require a knowledge of basic undergraduate mathematics (at least calculus and elementary linear algebra) and formal logic. Some class time will be devoted to background exposition of needed mathematics and physics. The course will sample a number of topics concerning the foundations and philosophical implications of modern physics. These will include: (1) probability and determinism in classical mechanics; (2) probability and indeterminism in quantum mechanics (and Bell’s Theorem); (3) “quantum logic”; (4) Euclidean and non-Euclidean geometry; (5) Minkowski spacetime geometry and the foundations of (special) relativity theory; (6) debate over the “conventionality of simultaneity”. For information contact D-malament@uchicago.edu.

Prerequisites: Advanced undergraduate course; also suitable for graduate students looking for a first course in the subject.

Texts: Readings on reserve in library TBA.

• **21-624 Selected topics: Differential equations, groups and Galois theory**

The fundamental question: “Why the integral \( \int e^{-x^2} \, dx \) cannot be expressed in terms of elementary functions?” lead Picard and Vessiot (in the 19th century) to develop an algebraic theory of ordinary differential equations that is analogous to classical Galois theory. This theory was continued in this century by Ritt and Kolchin. In the sixties the theory was connected to the theory of algebraic groups and number-theoretic questions. My interest in the field is motivated by some recent applications of model theory to arithmetical geometry (by Hrushovski) and the importance of the field to model theorists. The course will be an introduction to the basic classical results.

Prerequisites: No knowledge of logic is assumed, it is desirable that the students have had an introductory course in algebra covering Galois theory 21-610 or 21-374 (field theory).

• **21-621 Selected topics: Classical Descriptive Set Theory**

Please see [http://www.math.cmu.edu/users/eschimme](http://www.math.cmu.edu/users/eschimme)

• **21-700 Mathematical Logic II**

Introduction to higher-order logic (type theory) with primary emphasis on the typed lambda-calculus. Syntax and semantics, lambda-notation, axiomatic treatment, axioms of Descriptions, Choice and Infinity, weak completeness, weak compactness, standard and non-standard models; computer-assisted formal proofs; formalization of mathematics, definability of natural numbers, representability of recursive functions, Church’s Thesis; Gödel’s Incompleteness Theorems, undecidability, undefinability.

Prerequisite: An introductory logic course such as 21-600 or 80-610.

• **21-703 Model Theory II - The spectrum function**

Part of the course will be set in theoretic in nature: The fundamentals of Shelah “non-structure” theory will be presented, which are also the roots of proper forcing and pcf theory. The other part is structure theory (more like commutative algebra): The main results to be discussed are orthogonality calculus, and the structure of regular types. This will be used to prove to Shelah’s “Main Gap Theorem” for countable theories. If time will permit I will discuss also the solution to Morley’s conjecture (i.e., the mapping \( \lambda \mapsto I(\lambda, T) \) is a weakly monotonic). Probably will also make comments on the lower part of the spectrum function.

It is expected that the students know elementary set theory as well as the fundamentals of classification theory (including various equivalent properties to superstability, forking calculus, and the stability spectrum theorem).
Prerequisites: Little more than an elementary course in model theory (21-603) and a graduate course in algebra or permission of the instructor. The most basic parts of www.math.cmu.edu/rami/simple.dvi and www.math.cmu.edu/rami/DR.dvi.

Books: There is no official text. The following is the most relevant: Saharon Shelah, Classification Theory and the Number of Nonisomorphic Models, Rev. Eed., North Holland, 1990, Amsterdam.

• 80-705 Game Theory
The first part of the course will be a standard introduction to noncooperative games. The second part will cover special topics such as signalling and reputation effects, evolution and learning, formal models of players’ reasoning, and applications of game theory to artificial intelligence.

• 80-710 Recursion and Hierarchies
Alternate Years (Spring): 9-12 units An advanced course in computability theory and recursion theoretic hierarchies. Topics include finite-injury priority arguments, the arithmetical and Borel hierarchies, and the projective and analytical hierarchies.

• 15-711 Programming Languages
The phrase “mathematical logic” is ambiguous: one can interpret it as referring either to the mathematical study of reasoning, or to the study of mathematical reasoning. This course offers a proof-theoretic approach to mathematical logic, in both senses: we will use formal systems to model various aspects of mathematical proof, and then study these formal systems from a mathematical point of view.

Ideally, students taking this course should be familiar with the syntax and semantics of first-order logic (covered in 80-310/610), and the basics of computability and coding of syntax (covered in 80-311/611). Students should have taken at least one of the two courses just mentioned, or the equivalent.

The topics covered will likely be a subset of the following:

1. Pure logic: deductive systems, cut-elimination and applications, normalization, double-negation translations
2. The formalization of mathematics: e.g., number theory, analysis, and metamathematics in theories in first- and higher-order arithmetic and set theory
3. Finitary theories: primitive recursive arithmetic, $I\Sigma_1$, recursive comprehension, weak Konig’s lemma
4. Theories of arithmetic: Peano’s arithmetic, Heyting arithmetic, second-order arithmetic comprehension (and conservation over PwA, with speedup), realizability, and the Dialectica interpretation, ordinal analysis, the Paris-Harrington theorem.

• 80-711 Proof Theory
An introduction to proof theory that addresses philosophical, mathematical, and computational aspects of the subject. Topics may include: the emergence of Hilbert’s program, and proof theory’s more general reductive goals; deductive systems, double-negation translations, cut elimination, and normalization; the ordinal analysis of arithmetic; functional interpretation and the extraction of computational information from proofs; and subsystems of second-order arithmetic. Ideally, students taking this course should be familiar with the syntax and semantics of first-order logic (covered in 80-310/610), and the basics of computability and coding of syntax (covered in 80-311/611). Students should have taken at least one of the two courses just mentioned, or the equivalent.

• 80-712 Intuitionism & Constructive Mathematics
The course first presents basic principles of constructive mathematics (as formulated e.g. by Kronecker, Hilbert, Bishop) and the distinctive features of Brouwers intuitionistic analysis. Then intuitionistic logic and a variety of semantics are investigated. Finally, we discuss the fundamental difference between a constructivist and realist position in the foundations of mathematics.

• 80-713 Category Theory
Category theory, a branch of abstract algebra, has found many applications in mathematics, logic, and computer science. Like such fields as elementary logic and set theory, category theory provides a basic conceptual apparatus and a collection of formal methods useful for addressing certain kinds of commonly occurring formal and informal problems, particularly those involving structural and functional considerations. This course is
intended to acquaint students with these methods, and also to encourage them to reflect on the interrelations between category theory and the other basic formal disciplines.

Prerequisites: one course in logic or algebra

- **80-717 Constructive Logic**
  
  This multidisciplinary junior-level course is designed to provide a thorough introduction to modern constructive logic, its roots in philosophy, its numerous applications in computer science, and its mathematical properties. Some of the topics to be covered are intuitionistic logic, inductive definitions, functional programming, type theory, realizability, connections between classical and constructive logic, decidable classes.
  
  Prerequisites: 15-129 or equivalent and 15-212 for CS majors, a programming course and either 80-210 or 80-211 for Philosophy majors, 21-127 and one of 21-228, 21-484, 21-373, 21-132 for Mathematics majors.
  
  Comment: Design of this course is joint effort between Jeremy Avigad, Steve Awodey and Frank Pfenning with support by a grant for multidisciplinary course development from the University Education Council.

- **80-780 Linguistic Theory**
  
  At this point in time, linguistic theory is diverse, highly technical, and often rather abstruse. The goal of this course is not to develop technical ability in any one theoretical approach to language, but rather to reach an understanding of the philosophical and theoretical underpinnings of different approaches. We will begin by identifying the kinds of questions which linguistic theory seeks to answer, and will go on to examine the fundamental assumptions which underlie some of the theories which have been proposed to answer them. The course will be based primarily on readings from the linguistic and philosophical literature, through which we will examine foundational issues in generative grammar, functional grammar, and formal semantics. The course is appropriate for graduate students and for advanced undergraduates with a significant background in linguistics or in philosophy.

- **80-781 Formal Semantics**
  
  This course is designed to provide the student with a sound basis for intermediate and advanced work in the formal semantics of natural language. The first part of the course will review set theory, and propositional, predicate, and modal logic; and will introduce the student to intentional logic and categorial grammar. In parallel with the presentation of such formal systems, there will be a discussion of phenomena peculiar to the semantics of natural language, including the distinctions between sense and reference, direct and indirect quotation, propositional attitudes, generic and specific reference, quantifiers, and adverbials. The second part of the course will present one system for the semantics of English - Montague semantics - in great detail. Students should be comfortable with symbol manipulation and problem solving, and should have taken introductory logic courses before enrolling in this course.

- **21-800(A) Advanced Topics in Logic: Set Theory**
  
  This course is intended to be accessible to anyone who has taken the beginning graduate course “Introduction to Set Theory” or an equivalent course. I will assume knowledge of the rudiments of forcing and constructibility (essentially the material of Chapter I-VII in Kunen’s book).
  
  There will be little or no overlap with the content of my previous “Topics in Set Theory” course from Spring, 1998. The course will be divided into a series of three or four “minicourses” on different topics in set theory. The topics are negotiable and I welcome any reasonable suggestions. Some of the topics that I am considering are: An introduction to Shelah’s pcf theory and its combinatorial applications; The axiom of determinacy and its consequences; Forcing Axioms: from Martin’s Axiom to Marin’s Maximum; Proving consistency results from large cardinal hypotheses using forcing; Reflection and compactness phenomena in set theory.
  

- **21-800(B) Advanced Topics in Logic: Recursion Theory**
  
  This is a graduate recursion theory course with no prerequisite except a course in logic. Topics will include:
  
  - Effective computation and Church’s thesis
  - Hilbert’s 10th problem and other unsolvable problems
- Recursive functions and relations
- Recursively enumerable sets and relations
- The recursion theory
- Reducibilities among sets
- Post’s problem and finite injury priority proofs
- Degrees of solvability
- The lattice of RE sets
- The arithmetic, hyperarithmetic, and analytic hierarchy

Reference: Rogers’ Theory of Recursive Functions and Effective Computability.

- **21-803 Model Theory III**
  I will introduce Shelah’s notion of dependence in model theory, namely forking relation. It will be developed in the most general context in simple theories. It will be used to prove uniqueness of prime models for stable theories, thus answering a 30 year old conjecture of Ritt concerning the uniqueness of differential closure. The finite equivalence relation theorem, the stability spectrum theorem, unions of saturated models is saturated (under sufficient stability). Several extensions of these ideas will be discussed.

- **21-804 Mathematical Logic Seminar**
  The purpose of the seminar is to offer a forum for discussion of some recent developments in mathematical logic. We will concentrate in the areas of Model Theory and Set Theory. We will have weekly two hour meetings where faculty and students present papers (typically not authored by local people) at the level which will be appropriate to the participants (assume not much more than a first graduate course in mathematical logic). It is possible to take this seminar for credit, and in order to get a grade you will have to present at least one paper (usually 3 - 5 lectures). You may also be interested in participating passively (without giving a talk - not for credit). Every few weeks the topic will be changed, and often the subjects of discussion are independent.

- **15-810(A) Verification of Concurrent, Reactive, and Real-Time Programs**
  This course is a graduate level research seminar on automatic verification techniques for concurrent, reactive, and real-time programs. It was last taught in the fall of 1996, but different topics will be covered this semester.
  The prerequisites for the course are minimal-basic knowledge of elementary logic and automata theory. Students taking the class for graduate credit will be asked to prepare a short project and give one or two lectures. Auditors are also welcomed.

Logical errors in hardware controllers, communication protocols, and real-time programs are becoming an increasingly important problem. They can delay getting a new production the market or cause the failure of some critical device that is already in use. The most widely used method for verifying such systems is based on extensive simulation and can easily miss significant errors when the program is very complicated. Many of these programs can be viewed as having only a finite number of states. When this is the case, an alternative verification technique called “model checking” may be used. In this approach specifications are expressed by automata or temporal logic formulas, and programs are modeled as state transition systems. An efficient search procedure is used to determine automatically if the specifications are satisfied by the transition system. The technique has been used in the past to find subtle errors in a number of non-trivial designs. Recently, the size of the state transition systems that can be verified by model checking methods has increased dramatically. By representing transition relations implicitly using Binary Decision Diagrams (BBDs), it has become possible to check some examples with more than $10^{100}$ states!

**POSSIBLE TOPICS TO BE COVERED:**

1. Modeling concurrent programs with state transition systems
2. Temporal logics
3. The mu-calculus and fixpoint theory
4. The basic model checking algorithm
5. Binary decision graphs and symbolic model checking
6. Using Omega-automata to specify properties of concurrent systems
7. Using Omega-automata to specify properties of concurrent systems
8. Notions of equivalence for concurrent systems (observational equivalence, etc.)
9. Compositional reasoning techniques (e.g., the “assume-guarantee” paradigm)
10. Exploiting abstraction and symmetry
11. Using induction to reason about systems with many similar processes
12. True concurrency and models based on partial orders
13. Extending model checking techniques to handle real-time programs
14. Model checking techniques for the mu-calculus
15. Model checking for security and electronic commerce protocols.

• 80-810 Logic and Computation Seminar
This seminar is intended for Logic and Computation majors and beginning graduate students. It covers, on the one hand, a particular topic in logic, computation, or methodology; on the other hand, it introduces students to current work in the department with “guest lectures” by faculty members.

• 80-811 Thesis Seminar
This course provides a forum for the presentation and detailed discussion of research done by students, be they undergraduates working on their Senior Thesis or graduate students engaged with their M.S. Thesis.

• 15-812 Semantics of Programming Languages
This course introduces foundational concepts and techniques of programming language semantics. As a “starred” course it can be used by CS graduate students to fulfill their PL core requirement. We focus on two of the most successful styles of semantic description: denotational and operational semantics. We introduce semantic models chosen to be as simple as possible while supporting compositional (syntax-directed) reasoning about program behavior. We introduce techniques for reasoning about program correctness, for proving the correctness of a compiler, and we prove general laws of program equivalence. We apply these techniques to a variety of non-trivial examples.

Topics:

1. Introduction
   (a) denotational and operational semantics
   (b) inductive techniques for reasoning about programs
2. Sequential imperative programs
   (a) partial and total correctness
   (b) state transformation semantics
   (c) jumps and continuations
   (d) compiling to machine code
   (e) compiler optimizations
3. Parallel imperative programs
   (a) safety, liveness, and deadlock
   (b) fairness assumptions
   (c) shared-variable programs
   (d) communicating processes and dataflow networks
   (e) Kahn’s Principle for deterministic networks
   (f) trace semantics
4. Sequential and parallel functional programs
   (a) domains and continuous functions
   (b) recursion and fixed point theory
   (c) direct-style and continuation-style semantics
(d) call-by-name and call-by-value

Requirements: Grading will be based on homeworks, a midterm examination, and a final examination. Homeworks will be assigned regularly, with answers made available promptly.

References: Lecture Notes will be handed out on a regular basis.


- **80-812 Seminar on Causation**
  
  This course explores the foundations of causation. It examines how causal claims connect to both probability and to counterfactuals. Under a variety of background assumptions, and a variety of senses of "reliable", we will examine which causal inferences can be made reliably. We will also examine recent developments in statistics and artificial intelligence relating to causal inference.

- **80-813 Seminar on the Foundations of Mathematics**
  
  The seminar focuses on mathematical and logical work, important for foundation issues. That may range from the detailed presentations of “constructive” consistency proofs through conceptual analyses of central mathematical concepts to the discussion of ontological and epistemological issues.

- **15-814 Type Theory for Programming Languages.**
  
  This is an introductory course on the design and analysis of type systems for programming languages. The focus of the course is on the use of typed lambda-calculi and operational semantics as models of programming language concepts. These models are applicable to the design, analysis, and implementation of programming languages.

  PREREQUISITES: The course is appropriate for first- or second- year graduate students, especially those interested in further research in programming languages. Advanced undergraduates will be admitted of the instructor. Students are assumed to be familiar with ML and Java; programming assignments will be carried out in ML. Background in logic, semantics, and compilers is helpful, but not essential to the course.

  TEXTS: Readings will be drawn from the following sources.


  METHOD OF EVALUATION: Grades will be assigned on a pass/fail basis. Evaluation is based on class participation, bi-weekly homework assignments, and a final examination. Students are expected to participate in a note-taking cooperative.

  PROBABLE TOPICS: Inductive definitions; Static and dynamic semantics; Type safety; Function, product, and sum types; Universal types and polymorphism; Existential types and data abstraction; Recursive types; Object types; Subtyping, Equational reasoning; Type inference and unification.

  This course fulfills the core unit requirement for programming languages.

- **80-814 Seminar on the Philosophy of Science**
  
  A graduate level, critical review of standard issues in the philosophy of science. Topics will include determinism, predictability, confirmation, probability, causation, lawlikeness, explanation, the aims of science, the content of scientific claims, the rationality of belief in scientific claims.

- **15-815 Automated Theorem Proving**
  
  This course provides a thorough, hands-on introduction to automated theorem proving. It consists of a traditional lecture component and a joint project in which we will construct a theorem prover. The lecture component introduces the basic concepts and techniques of logic followed by successive refinement towards
more efficient implementations. The basic theorem proving paradigms we plan to cover are tactics, tableaux, and the inverse method, all three of which are applicable to classical and non-classical logics. Time permitting we may also cover some aspects of equational and inductive reasoning.

PREREQUISITES: Some exposure to ML would be helpful. Undergraduates should have taken 15-212/ML or 15-312.

TEXT: There is no textbook, but lecture notes and some papers will be distributed.

METHOD OF EVALUATION: The course grade will be based on 25% homework, 25% on a midterm, and 50% on the course project. There will be no final exam.

NOTE: due to the expected undergraduate participation, this class will start the first week of classes with introductory material on logic during the IC. Extensive notes on this material will be available to allow graduate students to join the class after the IC if they choose to.

TOPICS: Natural Deduction; Sequent Calculi; Unification; Tactics; Tableaux; Inverse Method; Term Indexing; Subsumption; Basic Paramodulation; Ordered Resolution; Induction.

- 80-815 Seminar on the Foundations of Statistics:
  The seminar focuses on some single important work, or body of work, and investigates it and related research from a contemporary point of view. For example, when Savage’s Foundations of Statistics is the course’s focus, the class goals include understanding how Bayesian decision theory differs from its rivals, and understanding where Savage’s position is located within the current Bayesian program. Other seminal thinkers whose writings have served as the course’s focus in different terms include, R.A. Fisher, Harold Jeffreys, J. Neyman, and A. Wald. Prerequisites: This is primarily a graduate level class. Instructor permission is required for undergraduates.

- 15-816 Linear Logic
  The course provides an introduction to linear logic with an emphasis on its applications in computer science. This includes the theory of functional, logic and imperative programming languages. We will also develop a linear type theory which will serve as a meta-language in which the theory of programming languages with state can be formalized effectively. We hope that an implementation of the type theory (currently in progress) will be available for practical experiments.
  TEXT: Course notes will be handed out.

  PREREQUISITES: General familiarity with functional programming would be helpful. Interested undergraduates are welcome, but require permission by the instructor.

  COMMENT: Course grades will be based on a combination of homework and a term project.

- 80-819 Seminar on Propositional Attitudes and Dynamic Semantics
  Bertrand Russell called a variety of verbs like ‘think’, ‘doubt’, ‘see’, and ‘hear’, propositional attitude verbs, because they seem to express some attitude that a person might have with respect to a proposition. Propositional attitudes provide an excellent test against the ontological and epistemological limitations of semantics. The seminar starts with a review of the treatment of propositional attitudes in the writings of Frege, the Oxfordian school of ‘ordinary language’, and some members of the post-Carnapian American tradition like Quine, Davidson, Kaplan and Kripke. Then we will focus on two recent semantic theories: Discourse Representation Theory (DRT) and Update Semantics. Although these theories seem to offer an ideal theoretical framework for propositional attitudes (due to their cognitive nature), they have been only recently applied in this field.

  The bulk of the seminar will be devoted to presenting the theories in a self-contained way, to comparing them with the approaches defended by Frege and Montague, and to considering how they can be applied to treat propositional attitudes. We will compare them also with other cognitive approaches in semantics. Although we might consider other attitudes in passing, we will focus on the analysis of belief and conditional belief.

- 15-819(A) Advanced Topics in Programming Languages: Domain Theory
  Course Description: The theory of domains has many applications in programming-language semantics and type theory, computability theory, and recently also in measure theory and chaos. The course will offer a basic foundational discussion of the idea building up to the solution of domain equations in suitable categories and constructions of special domains such as powerdomains and continuous algebras. A general theory of “equilogical spaces” will also be developed to provide an appropriate “realizability logic” of types derived from
domains. This course will be organized at a first-year graduate level, but is also suitable for seniors with a mathematical background. There will be weekly homework assignments. The final will be an individual take-home project to be assigned in the last third of the course. There will be no in-class examinations. Lecture notes will be provided and background reading supplied. There is no required textbook for the course. Prerequisites are reasonable “mathematical maturity” and some knowledge of computability (at, say, the level of the undergraduate FLAC course).

- **15-820/80-820 Seminar on Categorical Logic**
  This course focuses on applications of category theory in logic and computer science. A leading idea is functorial semantics, according to which a model of a logical theory is a set-valued functor on a category determined by the theory. This gives rise to a syntax-invariant notion of a theory and introduces many algebraic methods into logic, leading naturally to the universal and other general models that distinguish functorial from classical semantics. Such categorical models occur, for example, in denotational semantics, e.g., treating the lambda-calculus via the theory of Cartesian closed categories. Similarly, higher-order logic is treated categorically by the theory of topoi. Note: this course will begin with a 3 week refresher of basic category theory CS students can start after immigration by reviewing on their own.
  Prerequisite: 80-413/713 or equivalent

- **15-820 Seminar on Sheaves and Topos Theory** Seminar on topos theory and sheaf theory with applications to logic.
  Prerequisites: Category Theory 80-413/713 or equivalent.

- **15-827 Security and Cryptography**
  As use of the Internet has exploded, so has interest in security. This course will cover a range of topics on computer security including private and public key cryptography, authentication protocols, electronic commerce protocols, protocol design principles, logics of authentication, and security policy. More advanced topics include elliptic curve cryptography, zero-knowledge proofs, anonymous cash, electronic voting, electronic auctions, watermarking, trust management, and formal tools (such as model checkers and theorem proving) for reasoning about security. We will look at many of these topics from both a systems viewpoint, e.g., identifying typical classes of design flaws in authentication protocols, and a more formal viewpoint, e.g., verifying that an authentication protocol satisfies certain desirable properties. If time, we will look at code-level security issues such as addressed or raised by Java JDK, Active-X, Active Networks, software fault-isolation, and proof-carrying code.
  This course is intended to appeal to a wide range of students: people interested in state of the art cryptographic techniques, in designing new applied cryptographic protocols, in building systems using “off the shelf” security protocols, or in using mechanical tools for reasoning about security protocols. Students interested in pursuing research in any aspect of security are especially encouraged to enroll.
  Prerequisites: There are no formal prerequisites to this course but we assume students have had the equivalent of an undergraduate operating systems course, an undergraduate logic and discrete mathematics course, and some familiarity with distributed systems.
  Textbook and Readings: Handbook of Applied Cryptography by Menzes, Oorschot, Vanstone Papers distributed by instructors

- **15-850 Seminar on Linear Logic and Applications**
  This seminar examines linear logic with particular emphasis on applications in computer science. Basic topics to be covered include classical and intuitionistic linear logic, affine and relevance logics, natural deduction and sequent calculi, and decidability and complexity results for various fragments. Applications include linear type systems for functional languages, linear type inference, linear logic programming, concurrent languages based on linear logic, linear logical frameworks and type theories. Potential further topics include approaches to the model-theoretic semantics of linear logic and non-commutative linear logics.

- **15-850(B) Advanced Topics in Theory: Domain Theory**
  The theory of domains has many applications in programming-language semantics, computability theory, and recently in measure theory and chaos. The course will offer a foundational discussion of the idea building up to the solution of domain equations and constructions of powerdomains and continuous algebras. Then there will be a review of applications as time permits.
• 15-851 Computation and Deduction

We will explore the theory of programming languages using deductive systems. We will use such systems to specify, implement, and verify properties of functional, logic, and imperative programming languages. The deductive approach to the specification of programming languages has become standard practice, and one of the goals of this course is to provide a good working knowledge of how to engineer such language descriptions. Throughout the course we will use Elf (and an extension based on linear logic) as a uniform meta-language in which we can express specification, implementation, and meta-theory of the object languages we are considering. An implementation of Elf and examples will be available on-line for experimentation.

• 15-859(A) Complexity Theory

This is a graduate course the study of how much of a resource (such as time, space, parallelism, or randomness) is required to perform some of the computations that interest us the most. In a standard algorithms course, one concentrates on giving resource efficient methods to solve interesting problems. In this course, we concentrate on techniques that prove or suggest that there are no efficient methods to solve many important problems.

We will develop the theory of various complexity classes, such as P, NP, co-NP, PH, #P, PSPACE, NC, AC, L, NL, UP, RP, BPP, IP, and PCP. We will study techniques to classify problems according to our available taxonomy. By developing a subtle pattern of reductions between classes we will suggest an (as yet unproven!) picture of how by using limited amounts of various resources, we limit our computational power.

It has become increasingly obvious that proving even a small part of this world picture true (or false) would be a major mathematical achievement. For example, one part of this picture is that P does not equal NP. The question of the equality of these two classes was originally posed in a letter from Kurt Gödel to J. Von Neumann; it is one of the most important problem in all of mathematics. We will study some of the remarkable lower bounds which have been proved in the hopes of making progress on these central issues. We will see that the solution to these questions will require techniques unlike any that are currently known.

One especially interesting aspect of our study will be the variety of roles played by randomness. It helps in all aspects of complexity theory: finding efficient algorithms, finding reductions between problems, and proving that certain problems have no efficient algorithms in restricted models. Another point worth mentioning is that this seemingly pessimistic theory of what is impossible is what makes two highly practical fields of computer science possible at all: pseudorandom number generation and cryptography. These fields return the favor by suggesting what is true and what is provable in complexity theory.

Prerequisites: Mathematical maturity. Basic knowledge of some undergraduate mathematics: Linear Algebra, Probability, and Number Theory. For example, you should know: (1) what the basis of a vector space is, (2) that the expectation of random variable is additive, and (3) \( a^{p-1} = 3D1 \mod p \), for prime \( p \). Familiarity with the notion of algorithm and big O notation. Knowledge of Turing machines is helpful.

• 15-859(B) Computational Projective Geometry

Symbolic mathematical computation has developed to a stage where algebraic and numerical computation can be combined with computer-generated graphics to add dynamical depth to the investigation of many classical topics. This course will treat plane projective geometry in an algebraic implementation mechanized through the Mathematica symbolic computation software, with the objective to end up with the projective classification (over the complex numbers) of cubic curves.

Aside from presenting the necessary theoretical material, attention will be given to how symbolic algebra is applied to such topics. Previous familiarity with Mathematica is desirable, but special help sessions will be arranged for students not yet adept with the software.

There will be two one-and-a-half-hour lectures per week and two laboratory sessions.

• 15-885(A) Advanced Topics in AI/Computational Science: Computational Scientific Discovery (Crosslisted as 80-514/814 - Philosophy of Science Seminar)

This course surveys those tasks of scientific reasoning that have been automated in computer programs, the general methods that have been used, and what new analogous tasks in the sciences might be attempted. The emphasis is on high-end tasks: tasks that ordinarily are called creative.

As introductory background, we will review (1) characteristics of scientific research as carried out in practice, (2) some fundamental concepts and principles of scientific inference, and (3) methods of AI that have been most applied to scientific discovery.
Then, a series of case studies will be described from a variety of fields, including biology (cell & molecular), chemistry, physics, medicine, and social science. Discussions will center on (1) the context of the scientific task, (2) the computational method that is used, and (3) the role of prior knowledge and how it is handled. Finally, we will cover generalizations from these case studies, research methodology, the interface between scientist and discovery programs, and philosophical issues.

Course Requirements: If you are taking this course for credit, you will be expected to (1) do the regular homework assignments, which will include some hands-on experience with programs and maybe a little programming, (2) pass a midterm and final exam, and (3) keep a notebook and record questions that arise during your reading prior to class sessions. These questions will be raised and discussed during class.

Prerequisites: Graduate student status in SCS, Philosophy, MCS or consent of the instructor. Non-CS students should have some programming experience and appropriate undergraduate mathematics. No specific knowledge of the various sciences dealt with is needed.

TEXT:


(4) Other reading will consist of the primary literature and shorter selections from books.

Tentative Outline of Course:

Introduction (1 day)
Background
Textbook vs. Frontier Science (2 days)
Concepts of Scientific Inference (3 days)
Heuristic Search and Representations (3 days)
Scientific Discovery Programs
Historical modelling (2 days)
Early programs (2 days)
Midterm Examination (1 day)
Recent Programs (14 days)
Generalizations, Reserch Methods, etc. (3 days)
Final Examination

• 21-905 Seminar on Automated Theorem Proving

The seminar will probably meet once a week. The purpose of this seminar will be to help and encourage the participants to gain greater familiarity with certain parts of the literature on automated theorem proving, to become better acquainted with work on theorem proving at CMU and elsewhere, and to acquire the background on which future developments may be built. Choice of topics will be influenced by the interests of the participants.

In addition to 21-804, 80-810, 80-815, 80-819, 15-820, 15-850, and 21-905, the following graduate or general colloquia and seminars were conducted in recent years.

• Logic Colloquium

• Philosophy Colloquium

The above two colloquia are continuous and often present internationally renowned invited speakers.

• Seminar on Philosophy of Mathematics

• Principles of Programming Seminar

The POP Seminar is a weekly meeting of the Principles of Programming group, with presentations from its members or talks given by visitors.
Seminar in Logic and Computability I: Formal learning theory

In the past decade, the Carnegie Mellon Logic Community has organized the following conferences and workshops:

- Mid-Atlantic Mathematical Logic Seminar (February 90)
- Annual Meeting of the Association for Symbolic Logic (January 91)
- International Conference on Mathematical Foundations of Programming Semantics (March 91)
- Mid-Atlantic Mathematical Logic Seminar (April 93)
- Mid-Atlantic Mathematical Logic Seminar (March 95)
- Modern Mathematical Thought, Its History and Philosophy (Joint with the University of Pittsburgh) (September 95)
- Mid-Atlantic Mathematical Logic Seminar (April 97)
- Modern Mathematical Thought II: Historical and Philosophical Approaches (Joint with the University of Pittsburgh) (May 98)
- Graduate Conference in Logic & Philosophy of Science (Joint with the University of Pittsburgh) (March 99)
- International Conference on Automated Deduction (CADE-17) (June 2000)
- Midwest Philosophy of Mathematics Workshop (December 2000)
4 Degree and requirements

4.1 General requirements

The Program is an educational entity, not an independent academic unit in the university. Each student in the program will be enrolled in the Department of Mathematical Sciences, the Philosophy Department, or the Computer Science Department, and will be awarded a degree of Ph.D. in Pure and Applied Logic through this home academic department upon successful completion of the program.

Each student in the Program will follow a track corresponding to his or her home academic department. The degree requirements are generally determined by the tracks. All tracks require substantial course work in logic and logic-related subjects, and the completion of a Ph.D. thesis consisting of a scholarly exposition of substantial original research.

The courses required of each student will be determined by her or his advisor, within the guidelines of the track requirements. Among these may be courses offered by other Carnegie Mellon departments, or in some cases by other academic institutions.

The procedure for completing a thesis is determined by the regulations of the home college for each track. Work on the thesis will be supervised by the student’s advisor, and reviewed by the Logic Program faculty, as well as by the home department faculty. Progress on a thesis will follow the milestones and timetable customary for the home department. (For example, CS students will present a Thesis Proposal, produced under the supervision of the student’s advisor.)

On completion of a thesis, the student will present a Thesis Defense. The subject of this oral examination is the thesis and related topics. The examiners are usually, but not necessarily, the members of the Thesis Committee plus a non-departmental committee member as required by university regulations. This examination is public and must be announced throughout the university.

Students are also expected to informally participate in the activities of the Carnegie Mellon Logic community, such as relevant seminars and colloquia.
4.2 Track requirements and evaluation in Computer Science

The course requirements in the Computer Science track are:

1. Five star courses, one from each of these five areas:

2. At least five additional courses or core units in Logic, Logic-related areas, and other Theoretical Computer Science subjects. At least two of these should be offered by the Department of Computer Science. Since Computer Science Ph.D. students are restricted to taking the equivalent of at most one elective course (out of the required three) outside of SCS, in special cases, students may petition to have a second elective chosen from outside SCS.

Incoming students in the Computer Science track will participate in the Immigration Course. Students are expected to begin involvement in research during their first year, to complete their course requirements by the end of their second year, to satisfy the programming and communications skills requirements by the end of their third year, to present a Thesis Proposal during the third year, and to satisfy all degree requirements by the end of their fifth year. Students are expected to serve as teaching assistants for at least two semesters.

A student who has completed the course requirements, has completed the skills requirements, and has finished at least one of the two required teaching semesters may petition the faculty for a Master’s degree, which may be granted at the faculty’s discretion.

Computer Science/PAL students will be evaluated each semester at the regular Computer Science “Black Friday” meeting. Each student’s advisor will present an evaluation of the student’s progress at this meeting.
4.3 Track requirements and evaluation in Mathematical Sciences

The requirements for students in the Mathematical Sciences track (which are compatible with the requirements for other Mathematical Sciences graduate students) are:

1. A one or two-semester course in logic including proofs of the completeness and incompleteness, compactness, and undecidability theorems. Students may fulfill this requirement by taking Mathematical Logic I & II, or other suitable courses. An appropriate choice will be made in consultation with the student’s advisor. Of course, some students admitted to the program may already have fulfilled this requirement. They may still choose to take one or more of these courses to refresh or enhance their knowledge.

2. Additional courses in Logic and Logic-related areas. At least five courses shall be taken at Carnegie Mellon to fulfill requirements 1 and 2.

3. The following Mathematics courses are required:
   - 21-610 Algebra I
   - 21-620 Real Analysis (half-semester)
   - 21-621 Introduction to Lebesgue Integration (half-semester)
   - 21-640 Functional Analysis
   - 21-651 General Topology

4. The student will take additional courses to achieve a full course load of at least three courses per semester. Advanced students often sign up for Reading and Research.

5. Another requirement is the oral Qualifying Examination. breadth and depth of the student’s knowledge, especially in areas related to the student’s thesis research. The examination will have a major and a minor topic. The major topic will be the area of logic in which the student expects to do thesis research, and the minor topic should be a subject relevant to, but not closely related to, the research area. The minor topic may be chosen from the fields of mathematics, computer science, or philosophy, and need not be (but may be) a field of logic. The format of this examination is determined by the committee; it is common to have a brief presentation by the student at the beginning of the examination. The content of the examination must be agreed upon by the student and the committee not less than two months prior to the date it is scheduled and a document describing the content submitted to the Department. The examiners may choose to require the student to repeat all or part of the examination. An unsatisfactory performance on the second examination normally leads to dismissal. In exceptional cases the committee may choose to make part of the examination written rather than oral.
4.4 Track requirements and evaluation in Philosophy

The Philosophy Department at Carnegie Mellon University is distinguished by its precise approach to philosophical issues in:

Cognition, AI, and Philosophy of Mind

- history and philosophy of psychology
- artificial intelligence
- neural networks
- knowledge representation
- semantics and pragmatics of natural language
- foundations of computation

Decision and Rational Choice

- foundations of decision theory
- game theory
- rational choice
- political philosophy

Epistemology, Scientific Method, and Automated Discovery

- epistemology
- foundations of statistics
- belief revision and knowledge representation
- causal inference and discovery
- computational learning theory
- automated deduction

Logic and Mathematical Thought

- proof theory
- category theory
- constructive logic and type theories
- automated deduction
- logic of computation
- history of modern logic
- philosophy of mathematics
- philosophy of logic
The Ph.D. program seeks students interested in any of the above areas or combinations thereof and provides full tuition along with generous stipends for teaching assistance.

The department’s interdisciplinary research thrust affords an unusually broad range of career possibilities. Graduates of the program have been offered posts in Philosophy, Psychology, Computer Science, Statistics, and Industrial Research. Recent doctoral students supervised by CMU Philosophy faculty have taken jobs at the University of Alberta (tenure track, philosophy); the University of Washington (tenure track, statistics and tenure track, philosophy); the University of Texas (tenured, philosophy) and at Microsoft (Decision Systems Research Group). This wide range of interesting career opportunities reflects the department’s unique dedication to serious, interdisciplinary research ties.

Resources

The favorable institutional setting of the department, combined with its interdisciplinary ties, provides students with the opportunity to work on a broad range of applied projects unmatched in any philosophy department in the nation. The Ph.D. program is strengthened by coordination with the university’s nationally preeminent School of Computer Science and the Department of Mathematical Sciences. Faculty members hold joint appointments in such departments as Psychology, Social and Decision Sciences, Statistics, Mathematical Sciences, and Computer Science, and are active participants in such university research groups as the Center for Automated Learning and Discovery and the Human Computer Interaction Institute. Related resources in the Pittsburgh community include the departments of Philosophy, History and Philosophy of Science, and Medicine, as well as the Center for the Neural Basis of Cognition (run jointly by Pitt and CMU) and the Center for Philosophy of Science and the Archives for Scientific Philosophy (housed at the University of Pittsburgh).

Students benefit from the steady stream of visitors and speakers in the Department’s two colloquium series in Philosophy and Pure and Applied Logic. The second series is sponsored jointly with the departments of Computer Science and Mathematics.

The Curriculum

The core program is designed to be flexible, allowing students the freedom to develop programs emphasizing the various research strengths in the department. The following sample curricula illustrate how the core program may be extended in diverse directions reflecting the research strengths of the department.

Sample Curricula

- Cognition, AI, and Philosophy of Mind
- Decision and Rational Choice
- Epistemology, Scientific Method, and Automated Discovery
- Logic and Mathematical Thought
5 Admission and support

Applications for admission are directed to the Admission Committee for the Department of the applicant’s preferred track. Applications must be submitted on the forms supplied and include all the supporting material indicated in the application package. An application must clearly state interest in the Program in Pure and Applied Logic, and the Goal Statement should spell out and explain the candidate’s interests in and potential for doing research in Logic.

An application should be completed by Feb 1, including all records, GRE scores, recommendation letters and any other required material, to assure consideration by the track’s Admission Committee.

Only one track should be addressed as a first choice. Applicants who wish to be considered for a second track should address a letter to the Admission Committee of their second choice track and mention that fact in a cover letter with their main application. The second-choice Admission Committee will consider such applications concurrently with the main-choice Committee.

Admitted students are supported by their respective department, and earn this support as required by that academic unit. Students are treated as full fledged students within their home academic unit, notwithstanding possible variance of their degree and support requirements from those of fellow students in that unit.

Inquiries should be addressed to one of the following:

PhD Program in Pure and Applied Logic
Graduate Admission Committee
   Department of Mathematical Sciences
   OR
   Department of Philosophy
   OR
   Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3890
USA

E-mail (for any of the three departments) can be addressed to pal-program@cs.cmu.edu.