Computer Science 240 (Spring Term, 1997)

Computational Logic

Course instructor: Harry Mairson, 257 Volen (736-2724), office hours by arrangement. Send me *electronic mail* at address `mairson@lapis.cs.brandeis.edu` to arrange meetings, or to ask questions with the fastest, easiest response.

Teaching assistant: Anna Shubina 111 Volen (736-2728), electronic mail, `shubina@berry`.

Time and place: Tuesday and Friday, 10.30a.m.–12p.m., Volen 106.

Course Description

This course is a computer scientist's introduction to topics in logic that are relevant to computation, with applications to problems in theory (circuits and completeness theorems), software engineering (programming languages, data types, program synthesis), and artificial intelligence (formal planning). Unlike typical logic courses, it stresses computational metaphors: algorithmics, constructive mathematics, computational complexity, and decidability. The basic topics of the course are:

1. Propositional logic *(5 lectures)*.
2. First-order logic *(5 lectures)*.
3. Intuitionistic logic *(5 lectures)*.
4. Higher-order logic and realizability *(5 lectures)*.
5. Linear logic *(5 lectures)*.

The discussion of propositional logic includes a study in miniature of resolution-based theorem proving procedures, and of logical characterizations of complexity classes. The survey of first-order logic includes discussion of logic as a database query language, and a proof of the completeness theorem that can be used as a proof search technique (i.e., resolution). Intuitionistic logic introduces λ-calculus (the theory on which Lisp, Scheme, and other functional languges are based) as a mechanism for representing and manipulating proofs. Our discussion of higher-order logic stresses complexity issues, and how proofs can be analyzed to produce computational information, proving a version of the famous "Dialectica" theorem of Gödel. The course concludes with a discussion of *linear logic*, a so-called *resource-conscious* logic; we will discuss the relationship between cut-elimination (proof simplification) in linear logic, and evaluation of programs written in λ-calculus.

Required Work

Work for the course will include several problem sets, and possibly some class presentations.

Prerequisites

Some previous exposure to logic, computation theory, and functional programming. Suggested (but not required!) prerequisite courses are CS21b (Structure and Interpretation of Computer Programs), CS30a (Introduction to the Theory of Computation), CS140 (Logic Programming), and a course on logic (e.g., Math 125b/Philosophy 106b). As always, that good old "mathematical maturity" goes a long way. Needless to say, the course will be very theory-oriented, although I will try as hard as I can to make the material accessible, and to connect it to related topics in computer science.

Tentative syllabus (25 lectures plus introductory class)

0 Introduction and overview. *[This handout; Girard, Lafont, and Taylor:* 1.*]*

I: Propositional logic. *(5 lectures)*

I.1 Syntax. Truth tables. Natural deduction. Sequent calculus. *[van Dalen: 1.1, 1.2, 1.5; Girard, Lafont, and Taylor: 2, 5]*

I.2 Conjunctive normal form. Resolution theorem proving and completeness. *[Chang and Lee: 2.4, 5.1, 5.2, 5.7; class notes]*

I.3 Cut elimination. *[Girard, Lafont, and Taylor: 5, 13]*

I.4 Circuit value problem and PTIME-completeness. Satisfiability and NP-completeness. Quantified Boolean formulas and PSPACE-completeness. *[Hopcroft and Ullman: 13]*

I.5 Savitch's theorem (PSPACE=NPSPACE), Immerman's theorem (NPSPACE=co-NPSPACE). *[Hopcroft and Ullman: 13; Immerman]*

II: First-order logic. *(5 lectures)*

II.1 Syntax. Model theory (semantics). *[Chang and Lee: 3.1–3.2; van Dalen: 2.1–2.4]*

II.2 Natural deduction and sequent calculus. Undecidability of provability. *[van Dalen: 2.8; class notes]*

II.3 Datalog and query complexity: Linear queries and LOGSPACE-completeness; Nonlinear queries and PTIME-completeness. *[class notes]*

II.4 Completeness: Herbrand's theorem. Resolution theorem proving. *[Chang and Lee: 4–5]*

II.5 Boundedness and deductive databases. *[Hillebrand et al.]*

III: Intuitionistic logic. *(5 lectures)*

- **III.1** Foundations and philosophy of mathematics. Simply typed lambda calculus and Curry Howard isomorphism: programs as proofs. *[Girard, Lafont, and Taylor: 1, 2; van Dalen: 5.1]*
- **III.2** Strong normalization of proofs and proof search. Deduction theorem. *[class notes]*
- **III.3** Kripke models for propositional intuitionistic logic. Completeness theorem. *[van Dalen: 5.3; Troelstra and van Dalen: 2.5–2.6]*
- **III.4** "Double-negation" embedding of classical logic in intuitionistic logic. *[van Dalen: 5.2, pp. 172 ff.]*
- **III.5** PSPACE-completeness of intuitionistic theorem proving. *[Statman]*

IV: Higher-order logic and realizability. *(5 lectures)*

- **IV.1** Higher-order typed λ-calculi: systems F_1, F_2, F_3. Representing logical connectives. *[van Dalen, Pierce, Dietzen, Michaylov]*
- **IV.2** Representing and programming with inductive data types. *[Pierce, Dietzen, Michaylov]*
- **IV.3** Contracting proofs to programs. *[Leivant]*
- **IV.4** Metamathematics: proving termination and totality. A version of the Gödel Second Incompleteness Theorem. *[Leivant]*
- **IV.5** Decision problem for higher-order logic; deciding equivalence of simply-typed λ-terms. *[Mairson]*

V: Linear logic. *(5 lectures)*

- **V.1** Introduction to linear logic. *[Girard]*
- **V.2** Proof nets. *[Lafont]*
- **V.3** Coding linear logic and λ-calculus as proof nets. *[Lafont]*
- **V.4** Optimality and the geometry of interaction. *[class notes]*
- **V.5** Undecidability of derivability. *[class notes]*

References

Books:

Chin-Liang Chang and Richard Char-Tung Lee.
Symbolic Logic and Mechanical Theorem Proving.
Academic Press, 1973.
(chapters 2–5)

Dirk van Dalen.
Logic and Structure.
Springer-Verlag, 1983.

Jean-Yves Girard, Yves Lafont, Paul Taylor.
Proofs and Types.
Cambridge University Press, 1989.
(chapters 1–3, 5, 13)

John Hopcroft and Jeffrey Ullman.
Introduction to Automata Theory, Languages, and Computation.
Addison-Wesley, 1979.
(chapter 13)

Anne Troelstra and Dirk van Dalen.
Constructivism in Mathematics, Volume I.
North-Holland, 1988.
(chapter 2)

Christos Papadimitriou.
Computational Complexity.
Addison-Wesley, 1993.

Articles:

Jean-Yves Girard.
Linear Logic: Its Syntax and Semantics.
In **Advances in Linear Logic** (ed. J.-Y. Girard, Y. Lafont, and L. Regnier)
Cambridge University Press (1995), pp. 1–42.

Gerd Hillebrand, Paris Kanellakis, Harry Mairson, and Moshe Vardi.
Undecidable Boundedness Problems for Database Logic Programs.
J. Logic Programming 25:2 (1995), pp. 163–190.

Neil Immerman.
Nondeterministic space is closed under complementation.
SIAM J. Computing 17 (1988), pp. 935–938.

Yves Lafont.
From Proof Nets to Interaction Nets. In **Advances in Linear Logic** (ed. J.-Y. Girard, Y. Lafont, and L. Regnier)
Cambridge University Press (1995), pp. 225–248.

Harry Mairson.
A simple proof of a theorem of Statman.
Theoretical Computer Science 103 (1992), pp. 387–394.

Benjamin Pierce, Scott Dietzen, Spiro Michaylov.
Programming in Higher-Order Typed Lambda Calculi.
Computer Science Department, Carnegie Mellon University, 1989, Technical Report CMU-CS-89-111.

Richard Statman.
Intuitionistic Propositional Logic is Polynomial-space Complete.
Theoretical Computer Science 9 (1979), pp. 67–72.