My Experience Teaching Logic in Undergraduate AI at NYU

Ernest Davis
Dept. of Computer Science
New York University
New York, NY 10012
davise@cs.nyu.edu

May 31, 2022

Abstract

I have taught a section on logic in my introductory undergraduate and graduate courses in AI for many years. Here, I discuss what I teach; how students react; where else mathematical logic is taught at NYU; and what topics I would teach in a one- or two-semester advanced undergraduate course.

Regrettably, I only learned about the LogTeach-22 workshop on the day that submissions are due, so I don't have time to think through and write up a careful proposal. However, I have been teaching logic as part of introductory AI classes for forty years now, so I will describe my practice and experience and propose some ideas for a full course off the top of my head. Perhaps that will be interesting or useful to participants. I will not be able to attend the workshop.

What I teach

Since 1983, I have taught undergraduate AI fourteen times and the introductory master's level AI course (which is not actually very different) twenty-five times at New York University. Both courses are electives. The undergraduate course has a prerequisite of the algorithms class, which I waive for students with strong math backgrounds. Most of the students are seniors. Both courses currently get enrollments of almost 100 students per semester; ten years ago, the enrollment was about fifteen per year. The courses have always contained a component of logic; the content has changed somewhat over the years, but has been pretty fixed for the last fifteen years. Some of my course materials are on the open web[1] and I am happy to share problem sets and assignments.

In the undergraduate course, I spend three weeks out of fourteen on logic. Currently, the topics I cover are:

I. Propositional logic.

   A. Syntax and semantics (truth tables)
   B. Conversion to conjunctive normal form
   C. DPLL algorithm
   D. Compiling combinatorial and planning problems to SAT.

[1] https://cs.nyu.edu/~davise/AINotes.html

   E. Informal discussion of SotA SAT-solving and phase-transition effects.

II. Predicate calculus (without equality and with little use of function symbols)

   A. Syntax. (I don't discuss Tarskian semantics)
   B. Expressing facts in predicate calculus and avoiding common errors.
   C. Datalog inference

There are generally two problem sets on this material and one programming assignment. The programming assignment consists in implementing the DPLL algorithm, and writing a front-end that compiles some kind of limited planning problem (e.g. the peg jumping game; a toy adventure game with a maze and treasures) into SAT.

The graduate class covers all this material and, additionally, converting arbitrary sentences in the predicate calculus to CNF and the complete resolution theorem proving procedure for first-order logic. The assignments are the same.

Students' reactions

In my experience, on the whole, the students do well on this material, on both the assignments and on the exam. Practically all of them master the DPLL algorithm. (I always give a problem on DPLL on the midterm exam, and I can always be sure that 90% of them will do perfectly.) Converting combinatorial and planning problems to propositional logic and expressing propositions in predicate calculus are certainly more difficult; strong students master it, average students do well but can get tripped up on tricky examples, and weak students are lost. Overall, they find it less confusing than the material on foundations of probability theory, which is the next section of the course. How much they enjoy it, I can't say. I don't remember ever getting any particular complaints.

Where else mathematical logic is taught at NYU

The undergraduate *Discrete Math* course includes some material on propositional logic and naïve set theory. This is taught by the Math Department; it is a requirement of both the CS major and the Math major.

The undergraduate *Theory of Computation* course at NYU does not seem to contain any material on logic beyond propositional logic and the NP-completeness of SAT.

The department has on the books a graduate course entitled *Logic in Computer Science*, but this has not been taught in many years (since Clark Barrett took a job at Stanford.) The enrollment was small when it was taught.

There are graduate courses in abstract interpretation and formal methods. Professor Thomas Wies, who works and teaches in this area, tells me:

> I regularly spend an entire semester of our formal methods seminar on topics in logic to make sure that the PhD students in our group receive the relevant training. Last time we focused on first-order predicate logic (syntax/semantics, normal forms, compactness, Herbrand's theorem, resolution and refutational completeness, Goedel's incompleteness theorem, ...) and some basic proof theory (natural deduction, sequent calculus, Gentzen's Hauptsatz, ...).
>
> My graduate level course Rigorous Software Development covers Hoare Logic and some basic algorithms for automated reasoning in first-order logic: DPLL/CDCL, decision

procedures for various first-order theories that are relevant for reasoning about programs, theory combination, heuristics for reasoning about quantifiers, etc.

Some undergraduates have done research with Prof. Wies; those have gotten individual guidance from him.

The philosophy department offers two undergraduate courses on logic. *Logic* is "an introduction to the basic techniques of sentential and predicate logic." The *Logic* course is a prerequisite for the undergraduate philosophy major. *Advanced Logic* "tackle[s] a series of deep and beautiful results about the in-principle limits on what can be calculated, described, and proved by finite beings, culminating in Gödels incompleteness theorems." There are also undergraduate courses in *Set Theory* and *Modal Logic*, and an introductory graduate course *Logic for Philosophers*. My anecdotal sense is that a handful CS majors take the *Logic* course; very few take the more advanced courses.

It is worth noting that the otherwise distinguished math department at NYU does not teach a *single* course in mathematical logic at either the undergraduate or the graduate level. Undergraduates who want to learn the topic take the courses in the philosophy department, which is not ideal for math majors; graduate students presumably have to learn it by themselves. I raised this with the department a few years ago, but nothing came of it. NYU is certainly not unique in this respect.[2] It has not had any faculty primarily in the area since Martin Davis retired a couple of decades ago.

What I would teach in a one- or two-semester undergraduate course on Logic for Computer Science

If I were teaching a course on logic for computer science, my primary goal would be to teach students the foundations of the techniques used in AI reasoning, SAT solving, program verification, and mathematical proof verification. A secondary goal would be to give them a sense of the directions that formal logic has taken. So topics beyond the above might include:

- Advanced SAT-solving technology e.g. conflict clauses.
- First-order theories with equality and the associated inference techniques.
- Limited first-order theories e.g. Horn clause logic and Prolog
- Sorted logic.
- First-order logic with axiom schemas.
- ZFC axioms of set theory, and formalizing mathematical theories in set theory.
- Logic for databases.
- Closed-world and unique names assumptions.
- Higher-order logics.
- Discussion of verification systems like Isabelle and Coq.
- Temporal logics and dynamic logics.
- Axiomatizing real-world domains in formal logic (my personal area of research).

[2] As far as I can determine, no mathematics degree program at any level in the US requires a course in mathematical logic, despite the foundational nature of the subject. I think it is a safe bet that most research mathematicians have never taken a course in the subject.

- Tarskian semantics. Soundness and completeness.
- Sketch the proofs of Gödel's completeness and incompleteness theorems.