

Discovering Logic through Comics

Iliano Cervesato
Carnegie Mellon University — Qatar
iliano@cmu.edu

ABSTRACT

This paper describes a new experimental course introduced in the Spring of 2010 at Carnegie Mellon University in Qatar. *Discovering Logic* is an introduction to logic for Computer Science majors in their freshman year. It targets students who have had little or no exposure to logic and has the primary objective of 1) preparing them for sophomore classes which require proficiency with understanding formal statements expressed in English, elementary reasoning skills, and a sense of mathematical rigor. The course is structured as to achieve two additional objectives: 2) develop the students' communication skills, and 3) give them some historical depth into Computer Science and logic. This led to a somewhat unconventional approach that used a comic book, *Logicmix*, as the course textbook and that empowered the students to be active agents in the learning process through presentations and numerous open discussions. Preliminary analysis hints at an improved performance in follow-up courses, indicating that it may be achieving its primary objective.

Categories and Subject Descriptors

K.3.2 [Computers and Education]: Computer and information science education—*Computer science education*; F.4.1 [Mathematical Logic and Formal Languages]: Mathematical logic; K.2 [History of Computing]: Theory

General Terms

Experimentation, Human Factors.

Keywords

Logic, first-year of college, comic books, *Logicmix*, computational thinking, communication skills, history of CS.

1. INTRODUCTION

Computer Science draws significantly from logic [4, 3] and therefore CS students need to master basic logic and develop

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

June 27–29, 2011, ITiCSE'11 Darmstadt, Germany.
Copyright 2011 ACM 978-1-4503-0697-3/11/06 ...\$10.00.

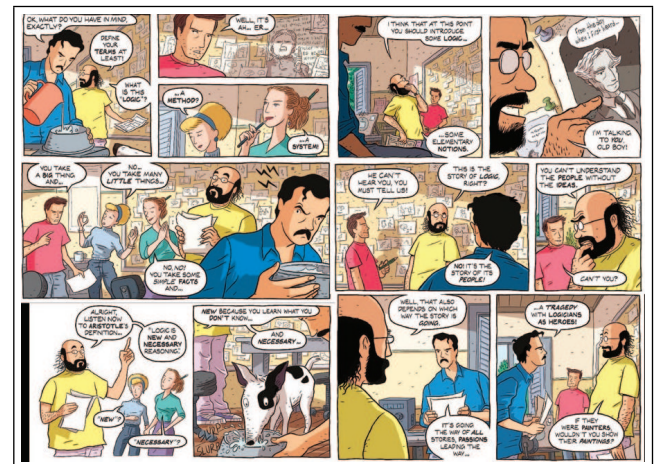


Figure 1: A sample page in *Logicmix* [5], where the authors portray themselves defining “logic”.

good logical skills to succeed in their major and in their career [1]. This means not only getting some acquaintance with propositional and predicate logic, but also having developed fundamental skills at reading informal texts and extracting the logic they intend to convey. This is necessary for example to apply known theorems or other techniques to solve a problem, to write a program from specifications, and in general to engage in any reasoning task. In our institution (and many like it), these elementary logical skills are taught over a mere couple of hours as part of a busy discrete mathematics course in the freshman year. There are indications that this is insufficient to prepare students for the demanding sophomore and upper-division courses.

As an experimental remedy, we have introduced a dedicated half-term freshman logic elective, *Discovering Logic*, whose primary goal is to bridge this gap. To make it more appealing to students, we made the unusual choice of using a comic book, the much acclaimed *Logicmix* [5], as its textbook. This provided the opportunity to expand the scope of the course to developing the students' communication skills through presentations and essays, and to giving them some historical depth into logic, mathematics and Computer Science. Altogether, this set a fun, down-to-earth, class atmosphere that kept the students engaged — an end-of-semester survey confirmed that the course was well received. We monitored the students' performance throughout the sophomore

year to assess whether it actually boosted their elementary logical aptitude compared with their peers who did not take the course — it apparently did.

This paper has two primary objectives: 1) to share an educational experiment with the community, and 2) to reflect on this experiment through the lens of recent research in CS education and related experiments.

This paper is organized as follows: we provide some context about logic in the CS curriculum and on the educational use of comic books in Section 2. We give a detailed description of *Discovering Logic* in Section 3 and of its learning objectives in Section 4. We report assessment results in Section 5.

2. BACKGROUND

We provide some context to the design of *Discovering Logic* by focusing on two characteristics of the course: the teaching of logic in the undergraduate CS curriculum and its use of comics in the classroom. We do not delve into other aspects, such as developing communication skills and student engagement, which deserve a much longer discussion.

2.1 Logic in CS

The ACM/IEEE 2001 Computing Curricula for Computer Science [1] and its 2008 interim revision [2] recommend that CS undergraduates be exposed to a minimum of 10 hours of “basic logic” (plus 12 hours on proof techniques) in their introductory courses, with additional logic taught as needed in courses such as databases, artificial intelligence and computability/complexity. These guidelines also leave the door open to students taking “advanced math electives such as logic” [1].

The initial exposure to logic is often confined to a busy discrete mathematics course in the freshman year which, at least in our program, is insufficient to provide students with the necessary logical skills for their subsequent CS courses. A fascinating recent survey of standalone introductory logic courses [12] indicates that they often mix a highly diverse population of students at all levels and from different majors (mainly philosophy). Based on an analysis of syllabi, they tend to focus on knowledge (definitions, properties, etc.) rather than skills (using logic as a learning tool, to solve problems, to engage in computational thinking).

Discovering Logic is designed to reinforce the discrete math exposure. It aims at building a solid foundation in those most elementary aspects of logic that are pervasively used throughout the CS curriculum: the basic skills that enable students to read a specification and translate it into a program, for example, or to read a problem statement and match it with known techniques to approach its solution. These skills quickly become second-nature, but acquiring them in the first place can be a struggle.

Many in the CS education community have advocated for more logic to be included in the CS curriculum [10, 3]. Paraphrasing [10] and [3], if logic is to Computer Science what calculus is to engineering, then it should be taught as such.¹ *Discovering Logic* is one concrete attempt to do so at the introductory level within the current system.

¹Interestingly, Myers’s passionate call for logic to take the place of calculus in CS is echoed in a milder way in [1, p.48] as “it is often more appropriate for computer science students to take less calculus and more courses in discrete mathematics or other material more directly relevant to the practice of computer science”.

2.2 Teaching with Comics

In the last 30 years, many science topics taught at the college level have been given new colorful forms as comic books: Computer Science was penciled out in this way as early as 1983 [7] and several new such books have been published in the last two years on subjects such as molecular biology [13], the theory of evolution [8] and now logic [5].

Surprisingly, we found little evidence of instructors using these and similar titles in their classes in the literature or on the Web, at least in science classes (comic books are occasionally used in K-12 humanities classes, and of course there are dedicated courses in graphic design curricula). The lone exception is Rota and Izquierdo’s use of a purpose-made comic book to teach plant biotechnology in Brazil [11]. One other related attempt we are aware of is Yim et al’s [14] use of cartoons hosted on the Web to illustrate CS concepts such as lists, etc. We strongly suspect however that more experiments have been attempted.

These graphic novels, comics and cartoons are an extreme form of the visual tools that are being used more and more frequently in teaching engineering and scientific subjects. In a recent multidisciplinary gathering [9], story telling and visual imagery were found to provide alternative ways to think about science and engineering, ways which promoted an effective form of visual learning. The benefits of visuals in engineering education were anticipated by Felder and Silverman [6] who observed that most people of college age have a predilection for visual information, while most college teaching has historically been verbal in these disciplines. The mentioned experiments narrow the gap between teaching and learning along this visual/verbal dimension.

3. COURSE ORGANIZATION

3.1 Origin and Motivations

Discovering Logic emerged from the confluence of a problem and an opportunity.

The *problem*, which had been pointed out repeatedly in our department over the years (but never really addressed), was that some students have difficulties working with formal statements in their sophomore year theory classes. Specifically, they have difficulties parsing English text and extracting the logical components (deep understanding) that allow them to solve problems and make the necessary connections to build up knowledge — this was observed both in the classroom, for example when students would have difficulties matching a theorem they knew well with a problem that paraphrased its hypotheses, and in exams or assignments where these parsing difficulties would cause them to spend a lot of time on a question. This problem was not universal, but a sufficient number of students manifested it as to elevate it to an issue of concern. Interestingly, once they extracted the logic, they rarely had difficulties applying formal rules or performing correct reasoning.

The *opportunity* was the publication in late 2009 of *Logicomix* [5], an intelligently crafted and beautifully illustrated comic book — the proper phrasing is “graphic novel” — which recounts the life of the logician Bertrand Russell as he witnessed and influenced the major developments of logic from the end of the 19th century to the eve of World War II. *Logicomix* is not a textbook, and indeed logic, although a pervasive theme, is never “taught” to the reader — as the authors say repeatedly, it is a story. The sample page of

Logicomix reproduced in Figure 1 exemplifies the rich multi-layered storytelling: there, it steps out of Russell’s narration of his life to a group of protesters and shows the book’s authors defining “logic”.

In the context of the course, *Logicomix* was used for several purposes, none of which being the teaching of the technical aspects of logic.

- The first purpose was motivational: a course using a comic book fueled interest. All students who enrolled got drawn into the book and quickly started asking questions about characters, events and concepts.
- The second purpose was to enable some of the secondary objectives of the course: the book exposes students to a great deal of the history of logic, from references to Aristotle to the visual acquaintance with Georg Cantor, David Hilbert, Kurt Gödel, and several other major logicians of the pivotal period in the development of logic the book is about, and even a preview of the imminent rise of Computer Science through brief encounters with John von Neumann and Alan Turing.
- It also provided starting points for the selection of presentation topics and fed some of the in-class discussion about more technical content.

3.2 Position within the CS Curriculum

Carnegie Mellon University in Qatar is a satellite campus of Pittsburgh-based Carnegie Mellon University, a medium-size research-oriented American university: we follow the same model, just on a smaller scale. Enrollment in the CS program fluctuates between 20 and 30 students a year. In the Spring 2010, the freshman cohort numbered 27.

Like in many CS programs that follow a liberal education model, students at Carnegie Mellon University are enrolled in just a sprinkling of Computer Science classes in their first year (mostly programming classes) with the bulk of their time spent in math and humanities. Students get exposed to more substantial CS courses in their sophomore year. The transition to more abstract and theoretical topics is significant for all students and brutal for the weakest performers. By and large, students are left to their own devices to cope with this transition.

CS students at Carnegie Mellon University receive a minimal exposure to logic in their freshman year: their introduction to logic consists of two 50-minute lectures (plus graded assignment problems) in a discrete mathematics course they take in their second semester, much less than the 10 hours recommended in [1, 2]. Their next three years in the program depend on these two hours: this is insufficient. Many students learn the logic they need along the way, but a few stumble. *Discovering Logic* is an attempt at helping freshmen build up their proficiency with elementary logic as they ascend towards their sophomore year.

3.3 Course Structure

We offered *Discovering Logic* for the first time in the second half of the Spring of 2010 as a 7-week mini. It was an elective and accounted for 3 units (which roughly translates into an expected workload of 6 hours per week). The class met once a week for 80 minutes. Starting with the third lecture, this time was divided into a student presentation part which consisted of two 15-minute presentations plus 5-minute questions and feedback, and a 40 minute lecture.

The course instructor has significant expertise in logic in CS having done research in this area for years.

Enrollment was restricted to CS freshmen in order to keep the course focused on their needs — we had to turn down requests from non-CS and from non-freshmen. Nine students enrolled in the class, which is rather high for an elective — one third of the freshmen cohort. Only one dropped the class towards the end, in spite of good performance

3.4 Course Contents

The course contents fell into two broad categories: core material on which the instructor lectured, and a number of side topics that the students introduced to their classmates.

3.4.1 Core Material

The core material covered a discussion of what logic is, a brief history and why it matters to Computer Science, as well as the motivation for using formal/symbolic languages. The course then turned to propositional logic, with an initial emphasis on expressing natural language statements into logical formulas (and vice versa), followed by the study of truth tables which introduced elementary notions of model theory, and by a gentle presentation of propositional natural deduction and the question of the equivalence between validity and derivability. The next module of the course followed the same script with first-order logic: it introduced predicates, variables and quantifiers, putting a lot of emphasis on expressing informal statements as formulas, truth tables were then upgraded to a simplified form of Tarskian models and the natural deduction rules for the quantifiers were introduced. Originally, the schedule planned for discussing a few more topics, in particular logical paradoxes and logic programming, but we ran out of time.

Logicomix motivated some in-class discussions, but otherwise played a minimal role as far as this core content is concerned. The students relied on handouts provided by the instructor and on the notes they took in class.

We chose a highly interactive style of instruction for this course. Each lecture was peppered with discussions, at first triggered mostly by the instructor and later erupting spontaneously. The class atmosphere was intentionally kept light-hearted and fun. Humor was encouraged. We typically introduced a topic by posing a question to the class and guiding the discussion through the use of examples and counterexamples until the underlying notion or principle was made explicit. For instance, the concept of valid inference was introduced by the example “Alice is in Kansas and Kansas is the US so Alice is in the US” which was modified back and forth until consensus emerged that for an inference of this type to be valid, it cannot be that its premises are true and yet its conclusion is false. As suggested in [1], such Socratic dialogs can be seen as a form of the scientific method as the students are asked to inductively analyze instances of a phenomenon to then deduce a general principle or formulate a definition. By following this approach, we moved toward bridging another of the dichotomies Felder and Silverman originally found in engineering education [6]: students tend to learn material presented inductively (from the concrete to the abstract) more easily while teaching at the college level is most often deductive (from principles to applications). We used this approach extensively in *Discovering Logic*, from the most elementary concepts such as defining “inference” to complex issues such as nailing down the natural deduction rules for disjunction and quantifiers.

3.4.2 Student Presentations

The side topics were the subject of student presentations. At the beginning of the course, we asked each of them to select a topic from one of three broad categories (people, logic, and others). The presentations had multiple purposes: expanding the scope of the class beyond the core contents, developing the students' ability to research topics and organize ideas autonomously, developing their presentation skills, and analyzing constructively each other's work.

Because we had gone through relatively little core material by then, early presentations focused on the life of actual logicians: one student made a superb presentation about Christos Papadimitriou (one of the authors of *Logicomix* and a renowned complexity theorist) while another told the class about George Boole. By then, the students had read *Logicomix* and the presentation topics shifted to the historical context: students gave talks on illuminism and positivism, on mathematics in the 19th century, and on dadaism. The last batch of presentations transitioned to logic topics not covered in class, namely paradoxes, non-Euclidean geometry, fuzzy logic, temporal logic, and theorem proving.

Because each presentation lasted 15 minutes, there was no time to cover these topics at any depth. Yet, the class got some exposure to aspects of logic that rarely make it in the CS curriculum (especially the historical/contextual presentations). Maybe more importantly, this activity provided the students with a substantial opportunity to develop their communication skills: they had to research a topic to a sufficient depth to tell a compelling story, they had to organize what they found to share their newly acquired knowledge and make this story interesting, and they had to stand up in front of a picky audience. At the beginning of a session, all students were given a rubric with which to evaluate and comment on the presentations in that session. Each presentation session was followed by a debriefing where, rubric in hand, the students in the audience pointed to what the presenter did well and what he/she could have done better.

By and large, the student presenters did an outstanding job. A couple had difficulties in the research phase, coming back to us with keywords they did not understand rather than with an intuition of what their topic was about — in one case this took numerous iterations to correct. The presentations went from a highly creative PowerPoint comic book to duller bullet points with a few pictures thrown in.

3.5 Coursework

The student evaluation relied on their participation in class discussions (15% of the final grade), on the presentations we just examined (30%), on five homework assignments (40%) and one final paper (15%).

The first assignment asked the students to read *Logicomix* and to write an essay about it in the form of a letter to a friend or a review for Amazon.com; it also asked for three questions that the student would like to see answered in the course. For the final paper, they read the book again and wrote a second essay about it in the light of what they had learned in the course. They were also presented with a selection of their best questions from the first homework. These two essays had the purpose to develop another aspect of the students' communication skills: writing and specifically critical writing. The work they submitted provided a window on the writing abilities they had developed in high-school and on their attitude towards writing as CS majors:

one student's essays were sublime, the majority were what was expected of freshmen, and only two students submitted subpar work, betraying that they had not yet reached the developmental stage of their peers.

The remaining four homework assignments consisted of exercises that let the students practice their understanding of the core contents of the course and broaden their perspective about it. Extracting logic formulas from natural language was a recurrent theme, which we tried to contextualize to drive other points home (for example, giving them sentences in Italian or in Chinese with the only knowledge of how the propositional connectives were written had the dual purpose of demonstrating that logic deals with symbols, not meaning). Other exercises explored the connection between logic and Computer Science: they were asked to write about controlling search in Google and beyond, to get "if" statements to do the right thing, to develop a hypothetical three-valued logic for non-terminating programs, to annotate small programs with assertions, and so on. A minority of the exercises were logical in nature: do derivations, check validity, find counterexamples, etc. Student performance varied from acceptable to quite good, which indicates that the level of these exercises was neither too high or too low.

3.6 Student Feedback

Throughout the course, we provided students with numerous opportunities to give us feedback. In particular, on the last day of class, students were given a 3-question written survey ("What is working?", "What can be done better?" and "Any other suggestions?"). The one unanimous complaint was that 80 minutes once a week was not enough. Highlights included the presentations, the homeworks and the course contents. Suggestions ranged from discussing more topics to having a play at the beginning of each class.

4. LEARNING OBJECTIVES

4.1 Learn Elements of Logic

In our current curriculum, core "basic logic" accounts for less than two hours of instruction, against the 10 recommended by [1]. The students who elected to take *Discovering Logic* added another 10 hours (7 hours of instruction time and 3 hours of presentations). Our course covers all the topics and all the learning objectives for "basic logic" recommended in [1, 2]. It also embraces the less formalistic approach advocated in [2], although at a very basic level.

4.2 Nurture Researching, Presenting, Writing

When faced with a new word or a puzzling explanation, some students turn to the Web or other resources to quench their curiosity. Others turn the page. Clearly, the first attitude is much preferable and we would like to cultivate it in our students: this is a form of active learning [6]. *Discovering Logic* actively encourages students to overcome inertia in two main ways. First, *Logicomix* is full of these new words and alluded descriptions, but also has a gripping story line (actually multiple) that stimulates curiosity — and graded essays about the book can be pretty stimulating too. Second, while putting together their presentation, the students had to go out of their way and carry out this kind of research.

Autonomous curiosity, as well as effective written and oral communication skills, are some of the professional practice that CS students are expected to have acquired by the time

they graduate [1]. This is further reinforced in [2]. These curricular guidelines state that “communication skills should not be seen as separate but should instead be fully incorporated into the computer science curriculum” [1, p.42]. Because they do not confine these skills to a standalone course (although many programs, like ours, do have a technical communication course), [1, 2] do not provide learning objectives for them. Yet they recommend that “students in computer science programs should be able to: communicate ideas effectively in written form; make effective oral presentations, both formally and informally; understand and offer constructive critiques of the presentations of other” [1, p.42]. *Discovering Logic* fully implements these recommendations.

4.3 Acquire Historical Depth

Students in physics, engineering and other experimental disciplines tend to be introduced to a good deal of historical anecdotes about their field. This is a lot less so in subjects like mathematics, logic and to a large extent Computer Science: ask a typical college student when limits came about and who defined the concept, and you are likely to get a blank stare back. We took advantage of *Logicmix*'s vivid description of the struggle of ideas that led to modern logic (and Computer Science) — the book's subtitle is *An Epic Search for Truth* — to help the students appreciate that many of the things they learn in their books, or that are now unquestioned in everyday life, took decades to be established, agreed upon, and accepted. “Proofs” are an example.

Here, *Discovering Logic* deviates from [1] which prescribes a single hour on the “history of computing” as part of the CS curriculum.

5. ASSESSMENT

As we said, the primary objective of the course was to give freshmen better logical skills in preparation for their more abstract sophomore courses. To assess whether *Discovering Logic* achieved this objective, we followed the entire freshman cohort through the first semester of their sophomore year and analyzed their academic performance. We focused on the one problem solving course, CS-251, that pervasively requires students to read formal statements and reason about them. We split the CS-251 students into those who took *Discovering Logic* (group A) and those who did not take it (group B). The idea was to compare both groups' average grade in CS-251 with the overall class average (group A+B). We tried to account for possible selection bias by first normalizing our data using each groups' performance in their first-semester freshmen CS and math classes (indeed, students who took *Discovering Logic* were on average better performers than students in group B — by normalizing, we canceled out this advantage). We then used this overall data to predict the CS-251 performance of the students in groups A and B and compared it with their actual performance in that course. The result is that the actual performance of the students in group A was 4% better than predicted (69.82 vs. 65.55) while the actual performance of the students in group B was slightly lower than predicted (60.77 vs. 60.93). Although encouraging, this analysis is not sufficient to conclude that *Discovering Logic* did achieve its primary goal: the sample sizes are too small for conclusive results (as indicated by other statistical tests we conducted). We plan to repeat this evaluation with future editions of the course.

6. CONCLUSIONS

In this paper, we have reported on a new experimental CS elective at Carnegie Mellon University in Qatar. *Discovering Logic* teaches basic logic at the freshman level while developing the students' communication skills and introducing them to the history of their discipline. The intentionally playful tone, engaging yet down-to-earth exercises, and the use of a comic book, *Logicmix* [5], kept the students motivated. Preliminary analysis indicates that students benefited academically from taking the course. We will repeat and expand *Discovering Logic* in the Spring of 2011, and continue monitoring the outcome.

7. REFERENCES

- [1] Computing Curricula 2001: Computer Science, Final Report. Joint Task Force on Computing Curricula, IEEE/CS, ACM, 2001.
- [2] Computer Science Curriculum 2008: An Interim Revision of CS 2001. Joint Task Force on Computing Curricula, IEEE/CS, ACM, 2008. Report from the Interim Review Task Force.
- [3] K. B. Bruce, P. G. Kolaitis, D. M. Leivant, and M. Y. Vardi. Panel: Logic in the Computer Science Curriculum. In *SIGCSE'98*, pages 376–337, Atlanta, GA, 1998. ACM.
- [4] M. Davis. Influences of Mathematical Logic on Computer Science. In *The universal Turing machine (2nd ed.): a half-century survey*, pages 289–299. Springer-Verlag, 1995.
- [5] A. Doxiadis and C. H. Papadimitriou. *Logicmix: An Epic Search for Truth*. Bloomsbury, 2009.
- [6] R. M. Felder and L. K. Silverman. Learning and Teaching Styles in Engineering Education. *Engineering Education*, 78(7):674–681, 1988.
- [7] L. Gonick. *The Cartoon Guide to Computer Science*. Barnes and Noble, 1983.
- [8] M. Keller and N. R. Fuller. *Charles Darwin's On the Origin of Species: A Graphic Adaptation*. Rodale Books, 2009.
- [9] M. B. McGrath and J. R. Brown. Visual Learning for Science and Engineering. Tech. report, ACM, 2006.
- [10] J. P. Myers. The Central Role of Mathematical Logic in Computer Science. *ACM SIGCSE Bulletin*, 22(1):22–26, 1990.
- [11] G. Rota and J. Izquierdo. “Comics” as a tool for teaching biotechnology in primary schools. *Electronic Journal of Biotechnology*, 6(2), 2003.
- [12] C. Schunn and M. Patchan. An Evaluation of Accelerated Learning in the CMU Open Learning Initiative Course “Logic & Proof”. Report, Learning Research and Development Center, University of Pittsburgh, May 2009.
- [13] M. Takemura. *The Manga Guide to Molecular Biology*. No Starch Press, 2009.
- [14] K. Yim, S. Ahn, and D. Garcia. Computer Science Illustrated. <http://csillustrated.berkeley.edu/>.