The Logic of Teaching Logic

Mark Engelberg

October 12, 2017

1 The Aim of Education

A modern approach to education recognizes that education is more than just a list of facts that we teach students to regurgitate. Especially in this day and age, when the Internet places nearly every bit of human knowledge at our fingertips, looking up facts is the easy part. In our rapidly-changing world, it is vital for students to be able to acquire new knowledge on their own and then reason effectively about this new information. Ultimately, our goal is to give students a solid grounding in problem solving and critical thinking skills.

2 The Heart of Humanities

Logic lies at the heart of problem solving and critical thinking. This has been known for literally thousands of years, ever since the ancient Greek philosophers such as Plato and Aristotle figured out that logic was the key that unlocks a precise way of reasoning about the world. And yet, where in the curriculum do we teach it? For the most part, sadly, we don't. We generally expect kids to absorb logic through osmosis, picking it up through some combination of disparate educational experiences, such as geometry proofs in math class, and literature analysis in English class.

With some luck, students may eventually encounter logic in a philosophy course in college, where logic is presented as a fundamental building block of human thought and culture. In that context, logic serves as a vital humanities subject: a tool to sharpen students' essays and arguments. It comes as no surprise that logic is a significant portion of the LSAT exam which assesses readiness for law school. Logic is arguably one of the most crucial skills for success in the liberal arts.

3 Where STEM Meets Critical Thinking

But it would be a mistake to think of logic as *solely* a humanities discipline. In 1847, George Boole's *Mathematical Analysis of Logic* defined logic formally and precisely, using the language of mathematics. This new treatment of logic revolutionized mathematics, giving mathematicians an unambiguous way to state and prove theorems, with proofs carefully composed of convincing chains of unassailable logical steps. Most people associate math most strongly with the arithmetic they learned in grade school, but modern mathematics is so much more than that. Thanks to the advent of Boolean logic (named after George Boole), mathematics is now the refined art of building abstract models, discovering deep truths, and crafting irrefutable proofs.

Mathematics, in turn, is a building block for all of the sciences, so the importance of logic soon spread to other fields. In the 1930s, Claude Shannon's thesis *A Symbolic Analysis of Relay and Switching Circuits*, demonstrated the application of Boolean logic to the design of electromechanical logic circuits, allowing computers to be built out of components called *logic gates*, processing all information as a pattern of ones and zeroes, trues and falses. And thus, the modern era of computing was born.

These days, computing intersects with just about everything, and one of the most valuable skills one can possess is the ability to build and leverage computational models. Computational modeling is therefore one of the most powerful tools in our modern problem-solving toolkit, whether you want to analyze DNA fragments, or design prescription drugs, or compose and analyze music, or simply make better business decisions than your competitors. *Computational thinking* is the new critical thinking skill for the modern world, or rather, it is a *set of skills* necessary to harness the power of computing to solve problems.

Computational thinking is valuable and important for all students, and so the integration of computational thinking into the K-12 curriculum has become a hot educational topic. What are all the facets of computational thinking, and when should we teach them? Many educators, computer scientists, and school administrators are working together to develop standards and answer these questions. But one thing is clear: since all of computing rests on a foundation of Boolean logic, an understanding of Boolean logic is one absolutely critical, fundamental computational thinking skill.

4 The Role of Logic in Programming

Let's take a closer look at the role of logic in programming. A program that always does the exact same thing regardless of input is not usually very interesting. What we want are programs that react in some way to the user. We can write such programs using conditional statements of the form, "**If** __________, **Then** do such-and-such." The programmer needs to fill in that blank with some sort of logical expression, a test that will evaluate to true or false. A robust understanding of logic is critical for filling in that blank correctly.

For example, maybe I want to say "**If** (the user clicks on the notifications icon **or** (the user has new messages **and** has **not** checked her messages in the past five minutes)), **then** alert the user about the new messages." One weakness of most introductory computer science curricula is that they fail to provide enough background in logic for students to write and understand these sorts of statements with precision. Many teachers simply advise their students to use their intuition about logical connectives such as **and**, **or**, and **not**, hoping that this intuition will be sufficient.

Unfortunately, there are several ways that our use of logical connectives in the

English language doesn't match the exact meaning of those terms as used in math and computer science. So, advising students to rely on their everyday experience in order to interpret logical expressions will often lead them astray. One glaring example is the logical connective **or**. In English, if I were to say something like, "Tonight I am watching a movie or am I eating pizza," you would probably be surprised if I did both. But in math and computer science, the connective **or** always allows for the possibility that both parts are true. This is known as the *inclusive or*.

Other problem spots for novice programmers involve grouping their logical subexpressions properly with parentheses, understanding negations (especially negations of compound logical expressions), and recognizing common subexpressions that can be factored out to simplify the overall expression. All of these become straightforward once a student has studied Boolean logic. For example, in terms of Boolean logic, we filled in the above blank with an expression of the form $P \vee (Q \wedge \neg R)$. In this form, one can easily figure out what happens when you **not** ($\neg$) the whole expression by applying De Morgan's laws, and easily transform this into a variety of equivalent expressions, some of which may make the program easier to understand and interpret.

Just as algebra gives us a powerful tool for manipulating and solving word problems by letting us express them with variables and arithmetic operators, *Boolean algebra* gives us a powerful tool for manipulating and solving questions about true/false statements by letting us express them with Boolean variables and logic operators. Boolean logic provides us with the mental model we need in order to have perfect clarity about how such true/false statements will be interpreted by the computer executing the program.

5 The Boolean Satisfiability Problem

We just looked at one concrete way that logic shows up in the day-to-day practice of coding. But Boolean logic is so fundamental to computer science, that it shows up again and again in a surprising number of contexts.

One of the deepest and most interesting areas of computer science is the study of something called "NP-complete" problems. NP-complete problems, broadly speaking, are puzzles whose solutions are easy to check, but hard to find. As a metaphor, imagine you have a combination lock. If I tell you the combination, it is easy for you to check — you just dial in the combination I told you and see if it works to open the lock. But if I don't tell you, the only decent strategy for finding the combination is to try every combination, one at a time, which is a slow and arduous process.

One of the most famous NP-complete problems is something called the *Boolean Satisfiability Problem*. The idea is that you have a big logic formula, and you're trying to assign truth values to the variables that will make the logic formula evaluate to true. Easy to check, hard to find. The "complete" part of NP-complete means that it is possible to transform any other problem from the same complexity class into an instance of the Boolean Satisfiability Problem (called SAT for short). Many of these transformations from other tough problems to SAT are known and well-understood, which makes SAT hugely important.

Computer scientists have consequently spent decades discovering and programming clever heuristics to solve SAT puzzles. The resulting program is called a SAT solver. It won't necessarily work on all SAT puzzles, but it works surprisingly well on the kinds of SAT puzzles that show up in the real world.

What all this means is that programmers with a mastery of Boolean logic have an astonishingly powerful weapon in their arsenal: for many of the world's toughest problems, it is possible to express the problem as a Boolean formula which must be true, using variables for the unknowns, and then pass the formula to a SAT solver to discover the values of the unknown variables.

The value of Boolean logic is acknowledged at the highest levels of the computer science community, for example, by legendary computer scientist Don Knuth, in his book *The Art of Computer Programming*. *The Art of Computer Programming* is an epic compendium of computer science knowledge begun by Knuth in 1962 and still being written and extended by him today. The book is his life's masterwork, and Knuth has devoted the most recent chapters of his famous book to Boolean algebra and the inner working of SAT solvers, emphasizing that these concepts should be tools in every programmer's toolbox.

6 The Value of Logic

When I was a kid, I had the rare opportunity to participate in a magnet math program that taught Boolean logic early, in the sixth grade. The experience was transformational, so as an adult, I wanted to offer the same kind of class for my new community. In the school district where I now lived, a logic course for kids was completely unprecedented, but I kept calling around to different schools in my area, volunteering myself as a logic teacher until I found a school excited about offering this to the students. And that's how, twenty years ago, I ended up teaching my first class — a mathematical logic course for middle school kids.

Years later, I heard back from several of those students from that first logic class. They reported the same kind of profound advantages that I had experienced as a result of my own childhood logic class: improvements in writing essays and organizing speeches, an improved aptitude for math and science, and a huge edge in programming. Some of these kids went on to become programmers, but not all of them. Some went on to pursue law, medicine and other disciplines — even those students reported significant benefits from learning logic early.

Formal logic is almost never offered at the middle school level, but when kids have the opportunity to learn it at a young age, the results are dramatic. Given the importance of logic, I've been dismayed by how little attention it gets within traditional K-12 math and programming curricula.

7 Robot Repair

As a puzzle inventor for ThinkFun, one of my dreams has been to develop a game that would bring early logic exposure to kids around the world, entirely through play. I finally had the opportunity to do that with a game called Robot Repair.

Robot Repair is the famous Boolean Satisfiability Problem from computer science, disguised and dressed up as a fun game for kids to play. Along the way, they learn the seven basic logic gates that form the basis for math, programming, and digital engineering. While solving the Robot Repair puzzles, kids discover for themselves many of the very same strategies that SAT solvers use to tackle the world's hardest problems. Kids who play Robot Repair will have an advantage that spans multiple domains.

8 Conclusion

Logic is one of the most valuable problem-solving and critical thinking skills that one can possess. Since the time of the ancient Greeks, logic has been the cornerstone of Western thought and philosophy, making it an indispensable skill for studying the humanities. Furthermore, mathematical Boolean logic, as defined in the mid-1800s, is the bedrock of STEM fields, particularly math and computer science. The cross-disciplinary impact of learning logic is nothing short of amazing. And yet, it is rarely taught, despite the recent focus on computational thinking, in which logic plays a key role. Let's work together to overcome this blind spot in our educational system, and get logic training into the lives of as many kids as possible.