Applied Logic - CS4860

Robert L. Constable
Cornell University

Abstract

Logic has been an important element of mathematics for over two millennia and of computer science (CS) since its birth. Logic is in fact one of the oldest academic disciplines, going back to Aristotle in 350 BCE and to Euclid in 300 BCE. Logic played a major role in shaping mathematics as can be seen in Euclid's *Elements* of Geometry and in the creation of *set theory* and *type theory* as languages for precisely expressing mathematical knowledge and for settling questions about its foundational meaning. One of the founders of CS is Alan Turing who received his PhD at Princeton University under the guidance of Alonzo Church, the first American mathematical logician. Church's famous *lambda calculus* was the direct inspiration and basis for the programming language Lisp developed by John McCarthy and subsequently for all of the functional languages that followed such as Standard ML, OCaml, Haskell, $F^\sharp$ and others.

The subject of *automated reasoning* was also developed in computer science starting with Newell, Shaw, and Simon. This line of research eventually led to *proof assistants* which are now important and widely used tools at the heart of computer science. They are used to precisely define the tasks that a software system is designed to accomplish and prove mathematically that the resulting code satisfies the precise specifications. One of the first modern proof assistants, Nuprl ("new pearl"), was built in CS at Cornell in 1984, it is still in use. Members of the Nuprl team went on to help create one of the most widely used proof assistants today, Coq. Recent research at Cornell has proved in Coq that the Nuprl logical rules are correct. This is another landmark result in logic and CS.

Proof assistants have created a new kind of logic just as chess playing computers such as Deep Blue created a new kind of chess. An unaided human does not stand a chance at winning a game of chess against Deep Blue nor even against the chess programs available on smart phones. On the other hand, the combination of a human and a computer *chess assistant* can beat Deep Blue and the "smart chess phones". Why is that and what is the lesson behind this fact? Humans can provide what chess champion Kasparov calls "*strategic guidance*." They can come up with creative ideas for how to win and find plans and moves that have never been tried before. Machines are much less good at this. The combination of human strategic guidance and the machines ability to rapidly try it out in detail by playing forward is a powerful partnership of humans and machines in playing games.

This example of a human-machine partnership can be significantly generalized. It applies to logicians, mathematicians, and computer scientists working with proof assistants to ensure that a software system will work correctly or that a mathematical result is "really true." The futurist Kevin Kelly says that we will be paid in the future by *how well we work with machines*. This course will broaden your horizons on how to work with a class of machine with a bright future, proof assistants, developing a subject with a very long distinguished history and a very bright future, applied logic.