A Practical Way of Teaching Logic to Computer Science BSc Students by Using State-of-the-Art Tools Including SMT Solvers

Gergely Kovásznai[a], Gabor Kusper[b], Csaba Biro[c]

[a]Eszterházy Károly University, Eger
kovasznai.gergely@uni-eszterhazy.hu

[b]Eszterhazy Karoly University
gkusper@aries.ektf.hu

[c]Eszterházy Károly University
birocs@aries.ektf.hu

Abstract

It is always a challenge to teach formal approaches such as logic to Computer Science BSc students. Sometimes they do not understand what makes logic a fundamental tool in the wide spectrum of computer science, especially in programming. Therefore, it seems vital to bring logic close to practice and programming as much as possible.

The Eszterházy Károly University renewed the curriculum of the subject "Logical Foundations of Computer Science" 2 years ago. This curriculum is divided into two parts that are dedicated to propositional logic and predicate logic, respectively. Regarding both kinds of logics, we put focus on (1) how to formalize real-life statements, (2) how to analyze semantical properties such as logical consequence, (3) how to translate logical formulas to the most common normal forms such as CNF, (4) how to apply certain algorithms such as resolution in order to analyze formulas, and (5) how to do this as efficient and automated as possible. Regarding propositional logic, the topic of propositional resolution is continued in the direction of SAT solving, covering DPLL and unit propagation only, but rather focusing on the practical usage of SAT solvers by introducing the DIMACS format and executing the SAT solver MiniSAT on concrete problems encoded in DIMACS. Regarding predicate logic, the topic of first-order resolution is continued in the direction of Prolog, only in a very introductory level, in order to make the students able to declare simple arithmetic problems, list operations and logic puzzles in the form of Prolog programs.

Our curriculum seems unique in that it involves an additional topic, Satisfiability Modulo Theories (SMT), in the middle of the curriculum. We think that the quantifier-free logic over linear integer arithmetic (QF_LIA) in SMT is able to act like a bridge between propositional logic and predicate logic for the students, since it provides higher-level tools than propositional logic does and it is not that high-level as predicate logic. Therefore students acquire knowledge on the SMT-LIB format, they learn how to declare certain logic and arithmetic problems, software verification problems and logic puzzles in SMT-LIB. We also show how to execute the state-of-the-art SMT solver Z3 on those problems.

We also developed a MOOC course for the subject "Logical Foundations of Computer Science" that implements this renewed curriculum.

Keywords: computer science, logic, education, SAT, SMT, Prolog.