How Hard is Bit-Precise Reasoning?∗

Gergely Kovásznai∗

∗Eszterhazy Karoly University, Eger, Hungary
kovasznai.gergely@uni-eszterhazy.hu

Abstract

Formal verification of hardware and software is important in the case
of critical systems, as shown by several accidents due to such problems as
overflow, rounding error, etc. Bit-precise reasoning over bit-vectors is a fun-
damental tool for attacking such verification problems. But how hard is
reasoning over bit-vector logics? It is essential to answer this question before
inventing solving approaches for bit-vector logics.

The talk gives an overview on the complexity of bit-vector logics [1]. Com-
plexity depends on what bit-vector operators are used and whether quantifiers
are permitted. For instance, the quantifier-free bit-vector logic (QF_BV)
with all the common operators is NEXPTIME-complete, which seems ex-
tremely high considering the fact that QF_BV is commonly used in hardware
verification. We will investigate how that complexity can be reduced to NP-
completeness or PSPACE-completeness by restricting the set of operators.

Quantifiers are often used in software verification problems such as termi-
nation analysis and invariant synthesis [2]. The bit-vector logic with quanti-
fiers and uninterpreted functions (UFBV) is 2-NEXPTIME-complete. Inter-
estingly, as it was proved recently, that logic without uninterpreted functions
(BV) is AEXP(poly)-complete [3].

References


Keywords: bit-precise reasoning, bit-vector, logic, complexity

MSC: 68Q17, 68M15, 68N30

∗Supported by the grant EFOP-3.6.1-16-2016-00001.