

Justifying All Differences Using Pseudo-Boolean Reasoning

Jan Elffers,^{1,2} Stephan Gocht,^{1,2} Ciaran McCreesh,³ Jakob Nordström^{2,1}

¹ Lund University, Lund, Sweden,

² University of Copenhagen, Copenhagen, Denmark

³ University of Glasgow, Glasgow, Scotland

{jan.elffers,stephan.gocht}@cs.lth.se, ciaran.mccreesh@glasgow.ac.uk,
jn@di.ku.dk

Constraint programming solvers employ intelligent reasoning and search algorithms to solve hard combinatorial problems in much better time than worst-case complexity analysis would suggest. As the algorithms involved get increasingly sophisticated, it becomes harder and harder to be sure that the implementations are correct. Unfortunately, neither software testing nor conventional verification techniques have been much help in addressing this.

An alternative approach, which is discussed in this talk, is to create a certifying algorithm. The idea is that alongside an answer, a solver also outputs a certificate that the answer is correct, and this certificate can then be verified by a much simpler external tool. A solution, on the one hand, is a certificate in itself, because it is only necessary to check that the solution satisfies all constraints. On the other hand, if there is no solution, a proof of this fact needs to be constructed on the fly, aka *proof logging*, because the solver can not remember all steps that were performed to conclude that no solution exists. Note that the certificate does not verify that the solver is correct, but it does tell us if a solver ever produces a spurious answer (even if it is due to hardware or compiler faults).

Proof logging is already well established in the Boolean Satisfiability (SAT) community, through a format known as DRAT. However, using DRAT to express more complicated reasoning appears to be infeasible in practice. Instead we developed a tool called VeriPB¹ which uses pseudo-Boolean reasoning and cutting planes proofs, together with reverse unit propagation, to give algorithm engineers an easy way of integrating powerful proof logging into existing code.

In this talk I will present our recent work published at AAAI-20² on pseudo-Boolean proof logging for justifying all-different reasoning in the context of constraint programming.

¹ <https://github.com/StephanGocht/VeriPB>

² Jan Elffers, Stephan Gocht, Ciaran McCreesh and Jakob Nordström. Justifying All Differences Using Pseudo-Boolean Reasoning. AAAI 2020.