The Potential of Interference-Based Proof Systems

Marijn J.H. Heule[1] and Benjamin Kiesl[2]

[1] Department of Computer Science, The University of Texas at Austin
[2] Institute of Information Systems, Vienna University of Technology

We want to encourage researchers to investigate the potential of proof systems that modify a given set of formulas (e.g., a set of clauses in propositional logic) in a way that preserves satisfiability but not necessarily logical equivalence. We call such modifications *interferences*, because they can change the models of a given set of formulas.

Interferences differ from classical *inferences*, which do not affect the models of a set of formulas, because they only allow the derivation of formulas (conclusions) that are implied by the original formulas (premises). Moreover, while inferences reason about the *presence* of formulas (the premises), interferences can be seen as reasoning about their *absence*. Most traditional proof systems such as Frege systems, sequent calculi, or resolution-based systems use conventional inference rules. Popular examples of these rules are the *modus ponens* (left) and the propositional *resolution rule* (right):

$$\frac{A \qquad A \to B}{B} \qquad\qquad \frac{C \vee l \qquad D \vee \neg l}{C \vee D}$$

The modus ponens allows to derive a formula B from two formulas A and $A \to B$. The resolution rule allows to derive a clause $C \vee D$ from two clauses $C \vee l$ and $D \vee \neg l$. In both cases, the conclusion is implied by the premises and it is only the presence of these premises that is relevant.

In contrast, a simple example for an interference rule is the rule of *blocked-clause addition* [4] for formulas in conjunctive normal form:

$$\frac{F}{F \wedge (C \vee l)}$$

The blocked-clause-addition rule allows to add a clause $C \vee l$ to a CNF formula F if, for every clause $D \vee \neg l \in F$, the resolvent $C \vee D$ of $C \vee l$ and $D \vee \neg l$ is a tautology. Here, the relevant precondition is the *absence* of any clause $D \vee \neg l$ for which $C \vee D$ is *not* a tautology. Extending the resolution calculus with this simple rule already leads to exponentially shorter proofs for many formulas.

Classical inferences are useful when formalizing mathematical reasoning and when the proofs themselves are objects of study. However, when automated-reasoning engines are used for more practical purposes such as software verification or the solution of optimization problems, the main requirement for proofs is that their correctness can be efficiently verified. This led to the development of highly efficient proof systems that are based on interference rules. The most popular of these systems are most likely the DRAT system [6], which is the current standard in SAT solving, and its generalization QRAT [3], which is the generalization of DRAT to QSAT solving (i.e., solving of the satisfiability problem of quantified Boolean formulas).

The usefulness of these two systems comes from their expressiveness. Traditionally, search-based solvers for SAT and QSAT produced resolution proofs. However, resolution calculi have their drawbacks because they cannot efficiently certify the wide range of techniques—like preprocessing, inprocessing, or symmetry-breaking [1]—employed by modern SAT and QSAT solvers.

In contrast, the interference-based proof systems DRAT and QRAT can express virtually all these techniques in a uniform manner. It therefore comes as no surprise that DRAT has become the standard for SAT solving. The system QRAT is still young and not very well understood, but it also shows great potential for the future.

Apart from these two systems, we just recently [2] introduced interference-based proof systems for propositional logic that are even stronger than DRAT. These systems, called *set-propagation-redundancy system* (SPR) and *propagation-redundancy system* (PR), allow for short proofs without the need for introducing new variables, which simplifies the proof search.

In interference-based proof systems, a proof is not just a sequence or tree of formulas where every formula is derived from earlier ones via inference rules; instead, a proof is a sequence of interferences that modify an existing formula (or a set of formulas) in certain ways that are guaranteed to preserve satisfiability or unsatisfiability. For instance, the DRAT system starts with a propositional formula in conjunctive normal form (i.e., a conjunction of clauses which themselves are disjunctions of literals) and modifies this formula by either adding or removing clauses via the following rules:

$$\frac{F}{F \wedge C}\ \text{(clause addition)} \qquad \frac{F \wedge C}{F}\ \text{(clause deletion)}$$

Here, the preconditions of the rules depend on whether a proof is supposed to be a proof of satisfiability or a proof of unsatisfiability.

In a proof of unsatisfiability, there is no precondition on the clause-deletion rule and the precondition for the clause-addition rule is that C must be a so-called RAT (short for *resolution asymmetric tautology*) with respect to F. At this point, it is not necessary to know exactly what a RAT is; the crucial thing is that a clause can be a RAT with respect to a formula without being implied by that formula. The RAT property is just an efficiently decidable criterion that ensures that F and $F \wedge C$ are satisfiability equivalent. In unsatisfiability proofs, the goal is to derive the *empty clause* which is, by definition, unsatisfiable. As every application of an interference rule preserves satisfiability, a derivation of the empty clause proves that the original formula must be unsatisfiable.

In a proof of satisfiability, the rule preconditions are the other way round compared to unsatisfiability proofs: there is no precondition on the clause-addition rule but in the clause-deletion rule, C must be a RAT with respect to F. Here, the goal is to derive the *empty formula* which is, by definition, satisfiable. Since all the applications of interference rules preserve *un*satisfiability, a derivation of the empty formula proves that the original formula must be satisfiable.

The following three properties of interference-based proof systems, which we will next discuss in more detail, are crucial because they are unusual when compared to traditional proof systems: (1) The application of interference rules does not only depend on the presence of certain premises but also on their absence. (2) Allowing the removal of certain formulas or subformulas (as with the clause-deletion rule in DRAT) can be beneficial when proving unsatisfiability. (3) Proofs can be used for certifying the unsatisfiability (or the validity) of formulas as well as for certifying their satisfiability (non-validity, respectively). We next discuss these points in more detail.

Reasoning about the absence of premises. Interference rules can be quite different from classical inference rules. In a classical inference rule, the presence of certain premises (like A and $A \to B$ in the case of the modus ponens) guarantees that a conclusion (B for the modus ponens) can be derived. Inference rules are therefore monotonic: Whenever a conclusion C can be derived from a set of premises F, it can also be derived from every superset of F. In contrast, interference rules can be non-monotonic as they only require that satisfiability or

unsatisfiability is preserved. A simple example for a non-monotonic interference rule is the above-mentioned blocked-clause-addition rule: Assume that $C \vee l$ could be added to a formula F using the blocked-clause-addition rule. If, before adding $C \vee l$ to F, we first add a clause $D \vee \neg l$ for which $C \vee D$ is not a tautology, then the addition of $C \vee l$ is not allowed anymore. Other examples for non-monotonic interference rules are the addition of RATs and the *extension rule* [5] (which allows to introduce new definitions) in propositional logic, or the QRAT addition rule [3] for quantified Boolean formulas.

Allowing the removal of formulas. There are two main reasons why the removal of formulas can be beneficial. The first reason is that it can simply shrink the set of formulas under consideration so that automated proof checkers require less memory when verifying the correctness of a proof. The second reason has to do with the non-monotonicity of interference rules: The removal of formulas can enable the application of certain interference rules that could not be applied before. Consider as a simple example again the blocked-literal addition rule: It could be that the rule cannot be used for adding a new clause $C \vee l$ because there exists a single clause $D \vee \neg l$ such that the resolvent $C \vee D$ is not a tautology. In this case, the removal of $D \vee \neg l$ can enable the application of the blocked-clause-addition rule. It can actually be shown that the use of clause removal can lead to increased expressivity: In a yet unpublished result, we have proved that a certain QBF calculus (a subset of QRAT) becomes exponentially stronger when we extend it with an unrestricted clause-deletion rule.

Proofs for satisfiability and unsatisfiability. As we have already seen on the example of the DRAT system, interference-based proof systems can be used both for showing the validity or unsatisfiability of a set of formulas as well as for showing that some set of formulas is *not* valid or *not* unsatisfiable (i.e, satisfiable). This is especially interesting for formalisms such as first-order logic, where certificates of satisfiability cannot be represented as simple truth assignments and are often hard to verify. For instance, if a saturation procedure in first-order logic produces a saturated set and therefore concludes that the input formula is satisfiable, then one cannot simply extract a model of the formula from this set. Moreover, the set is usually only saturated modulo clause redundancy, which makes it hard to check whether the set at hand is indeed saturated. Because of this, a uniform proof format that can be used for certifying the satisfiability of many formulas would be beneficial.

We have seen that there exist promising interference-based proof systems for automated reasoning and that such proof systems can differ significantly from traditional inference-based proof systems. At this point, these proof systems are not yet well understood, so many questions for the research community remain: What is the potential of such proof systems? Could the use of interference rules lead to useful proof systems for formalisms like first-order logic, modal logics, separation logics, or others? What are the strengths and weaknesses of these proof systems? Can such systems lead to better methods for showing satisfiability of formulas, for instance in first-order logic? With this paper, we want to motivate researchers in automated reasoning to take a closer look at interference-based proof systems and to find answers to these open questions.

References

[1] James Crawford, Matthew Ginsberg, Eugene Luks, and Amitabha Roy. Symmetry-breaking predicates for search problems. In *Proc. of the 5th Int. Conference on Principles of Knowledge Repre-*

sentation and Reasoning (*KR 1996*), pages 148–159. Morgan Kaufmann, 1996.

[2] Marijn J. H. Heule, Benjamin Kiesl, and Armin Biere. Short proofs without new variables. In *Proc. of the 26th Int. Conference on Automated Deduction* (*CADE-26*), *Accepted*, LNCS. Springer, 2017.

[3] Marijn J. H. Heule, Martina Seidl, and Armin Biere. Solution validation and extraction for QBF preprocessing. *Journal of Automated Reasoning*, pages 1–29, 2016.

[4] Oliver Kullmann. On a generalization of extended resolution. *Discrete Applied Mathematics*, 96-97:149–176, 1999.

[5] G. S. Tseitin. On the complexity of derivation in propositional calculus. *Studies in Mathematics and Mathematical Logic*, 2:115–125, 1968.

[6] Nathan Wetzler, Marijn J. H. Heule, and Warren A. Hunt Jr. DRAT-trim: Efficient checking and trimming using expressive clausal proofs. In *Proc. of the 17th Int. Conference on Theory and Applications of Satisfiability Testing* (*SAT 2014*), volume 8561 of *LNCS*, pages 422–429, Cham, 2014. Springer.