# A General Nogood-Learning Framework for Pseudo-Boolean Multi-Valued SAT

## (Extended Abstract*)

Siddhartha Jain[1], Ashish Sabharwal[2], and Meinolf Sellmann[2]

[1] Brown University, Dept. of Computer Science, Providence, RI 02912, USA
sj10@cs.brown.edu
[2] IBM Watson Research Center, Yorktown Heights, NY 10598, USA
{ashish.sabharwal,meinolf}@us.ibm.com

Jain et al. [4] recently introduced a new no-good learning approach for multi-valued satisfaction (MV-SAT) problems. This approach was shown to infer significantly stronger no-goods than those inferred by a mechanism that is based on a Boolean representation of a multi-valued problem as a SAT instance. Like earlier methods, the learning approach is based on an implication graph where nodes represent domain events and edges are implications drawn by the clauses in the given problem. One of the novelties of Jain et al. was the sole focus on variable *inequations* to infer the minimal reasons for a failure.

In this work, we investigate why the particular use of inequations results in stronger no-goods and we formulate a general framework for multi-valued nogood-learning that can handle more general constraints, and also different domain representations, such as interval domains, which are commonly used for bounds consistency in constraint programming (CP). This is an essential step towards an integration of pseudo-Boolean and multi-valued SAT.

State-of-the-art SAT and CP methods differ significantly in style and philosophy, in particular in the problem representation as a "flat" set of clauses for the former and with richer multi-valued constraints (such as linear inequalities, element constraints, all-different, knapsack, etc.) for the latter. Despite this inherent structural advantage that CP offers, CP solvers that are based on SAT reformulations are highly competitive, such as `Sugar` [7], which won the CP global constraints competition in two consecutive years [2]. Consequently, various hybrid approaches have been introduced, including pseudo-Boolean solvers (e.g., Dixon et al. [3]) supporting inequalities over binary variables, SAT modulo theories (SMT) [5] "attaching" additional theories to a SAT solver, and the award-winning lazy clause generation approach [6]. In the adjacent field of mixed integer programming (MIP), conflict-driven nogood learning has been explored by the SCIP solver [1]. Another related contribution is the recently proposed multi-valued SAT solver `CMVSAT-1` [4] that strengthens nogood learning by directly incorporating the knowledge that each variable must take exactly one out of a number of values.

* Full paper to appear at AAAI-2011, the 25th Conference on Artificial Intelligence, San Francisco, CA, August 2011.

In this work, we provide a framework for strong multi-valued nogood learning, generalizing CMVSAT-1. We identify sufficient conditions for constraint propagators to support nogood learning effectively. Moreover, we show that the generalized framework is capable of handling various domain representations, such as interval domains, and associated propagators that enforce bounds consistency.

In CP, one may view the set of constraints as split in two classes. *Primitive constraints* are enforced directly by the domain representation used. *Secondary constraints* consist of the rest. The *propagation* of a secondary constraint $C_s$ can be viewed as the process of entailment of new primitive constraints $C_p$ from the conjunction of the existing primitive constraints and the secondary constraint that is propagated, i.e., $(\bigwedge_{C_i \in \text{PrimStore}} C_i) \wedge C_s \vdash C_p$.

In our framework, nogoods take the form of *disjunctions of negated primitive constraints*. The idea is to create an implication graph whose nodes represent primitive constraints. The outgoing arcs link a node to a set of primitive constraints that, when conjoined with some secondary constraint, allow the entailment of the corresponding primitive constraint. A conflict is reached when a subset of nodes in the graph represents a set of primitive constraints that contradict one another. A "valid" cut in this conflict graph (defined appropriately) provides us with a nogood that can be inferred from this conflict. We show that under certain conditions, we can efficiently learn powerful nogoods using such an analysis. Further, we propose a way to post-process the implication graph in order to further strengthened the learned nogood in this general setting.

We demonstrate the effectiveness of these ideas in an empirical evaluation involving problems from three domains, each combining challenging combinatorial constraints with linear inequalities: the 'quasigroup with holes' problem with costs, the market split problem, and the weighted N-queens problem. Our implementation of the above ideas as the solver CMV-SAT2 turns out to be the most robust method that works well across all these domains, compared to the pure SAT solver MiniSAT, the CSP solver Mistral, and the MIP solver SCIP.

## References


1. T. Achterberg. SCIP - a framework to integrate Constraint and Mixed Integer Programming. Technical report, 2004.
2. CSP Competition. International CSP competition result pages, June 2008-2009.
3. H. E. Dixon, M. L. Ginsberg, and A. J. Parkes. Generalizing Boolean satisfiability I: Background and survey of existing work. *JAIR*, 21:193–243, 2004.
4. S. Jain, E. O'Mahony, and M. Sellmann. A complete multi-valued sat solver. In *CP-2010*, pages 281–296, St. Andrews, Scotland, 2010.
5. R. Nieuwenhuis, A. Oliveras, and C. Tinelli. Solving SAT and SAT Modulo Theories: From an abstract Davis-Putnam-Logemann-Loveland procedure to DPLL(T). *J. ACM*, 53(6):937–977, 2006.
6. O. Ohrimenko, P. J. Stuckey, and M. Codish. Propagation = lazy clause generation. In *CP-07*. Springer-Verlag, 2007.
7. N. Tamura, T. Tanjo, and M. Banbara. System description of a SAT-based CSP solver Sugar. *Proceedings of the Third International CSP Solver Competition*, pages 71–75, 2008.