Using SAT-Solvers to Compute Inference-Proof Database Instances (Abstract Version)

Cornelia Tadros and Lena Wiese

Technische Universität Dortmund, 44221 Dortmund, Germany
{tadros,wiese}@ls6.cs.uni-dortmund.de
http://ls6-www.cs.tu-dortmund.de/issi/

Controlled Query Evaluation (CQE) is a logical framework that provides a basis for inference control in database systems. In [2] a preprocessing procedure (which we call *pre*CQE here) is described that accepts propositional input. The reason why we confine ourselves to propositional logic is that we can use up-to-date SAT solver programs for the computation of *pre*CQE solution instances.

In [2] it is shown that with certain system settings, the problem of finding an inference-proof instance db' amounts to finding a model $I^{db'}$ (hence, a satisfying interpretation) for a constraint set C.
To meet the availability requirements and thus retain as much correct information in db' as possible, we define two distance measures: the first one to measure how many entries of an explicit availability policy are affected by distortion and the second one to measure how many entries of the original database entries are affected by distortion: Note that, due to the model requirement, inference-proofness and hence confidentiality of the secrets is our main goal and the two distance measures are availability optimization functions.

The Branch and Bound approach for propositional logic in [2] can be encoded by a transformation of the input constraints such that the distance value need not be maintained explicitly. More precisely, *pre*CQE for propositional logic can be seen as a variant of an optimization problem for the satisfiability (SAT) problem. In the following we present the representation of the *pre*CQE problem as a weighted partial MAXSAT (W-PMSAT) optimization problem. Here it is crucial to see the constraints C as a set of clauses. Each clause has an associated non-negative integer as a weight. The optimization function is to maximize the sum of weights of satisfied clauses in an interpretation. Some clauses (those with a weight above a predetermined threshold) are explicitly designated as "hard constraints" that necessarily have to be satisfied; that is why the optimization is partial: the W-PMSAT solver only has to maximize the summed weight of satisfied "soft constraints". We can show that a solu-

tion of this W-PMSAT input represents an inference-proof, availability-preserving and distortion-minimal propositional solution instance for the *pre*CQE input.

In recent years, propositional SAT solving has seen a huge improvement in performance. Several highly efficient implementations take part in the yearly SAT competition (in conjunction with the SAT conference). As part of the SAT competition there also is a "MAXSAT evaluation" [3, 1] that includes competition categories for W-PMSAT problems. Those SAT solvers often employ a Branch and Bound strategy for propositional input (similar to the one described in [2]) and beyond that implement highly efficient heuristics to speed up the search. While the SAT competition is already quite established, the MAXSAT evaluation has been organized just for the fourth time in 2009. This shows that the interest in efficient solving strategies for this optimization problem has come up very recently.

We wanted to apply this highly efficient W-PMSAT technology to our problem and benefit from up-to-date solver implementations. To this end, we developed a program that translates propositional *pre*CQE input formulas into a W-PMSAT instance.
To test our prototype we made an effort to simulate problems specific to the database domain. As the tests were run with differently sized inputs, for every input size we tested 10 randomly permuted instances to avoid a bias caused by the input order.

References

1. Josep Argelich, Chu Min Li, Felip Manyà, and Jordi Planes. MaxSAT evaluation. `http://www.maxsat.udl.cat/`.
2. Joachim Biskup and Lena Wiese. Preprocessing for controlled query evaluation with availability policy. *Journal of Computer Security*, 16(4):477–494, 2008.
3. Federico Heras, Javier Larrosa, Simon de Givry, and Thomas Schiex. 2006 and 2007 Max-SAT Evaluations: Contributed Instances. *Journal on Satisfiability, Boolean Modeling and Computation*, 4(1):239–250, 2008.