Lingeling and Friends at the SAT Competition 2011

Armin Biere

Institute for Formal Models and Verification
Johannes Kepler University, Linz, Austria

Abstract. This note serves as system description for our solvers submitted to the various tracks of the SAT Competition 2011 affiliated to the SAT conference 2011.

Overview

To the main track of the SAT Competition 2011 we submitted the new version 587 of our SAT solver Lingeling and its parallel extension Plingeling [1]. To the minimally unsatisfiable subset track we submitted front-ends of PicoSAT version 941, one for each of the sub-tracks.

The plain MUS front-end PicoMUS uses multiple rounds of randomized clausal core extraction as preprocessing step before switching to selector variable based core minimization.

The group-oriented MUS extractor PicoGCNF only uses the latter technique, which is actually available as new API function in the PicoSAT library and computes a minimal subset of failed assumptions for assumption based incrememental SAT solving.

In order to use the clausal core feature, PicoSAT is compiled with proof tracing support, which is expected to slow down plain solver speed. Thus, for comparison, we submitted this version of PicoSAT also to the main track of the competition.

To the MiniSAT hack track we submitted a version of MiniSAT 2.2.0 which uses our agility metric [3] to control when to disable backtracking during restarts with agility limit 26%.

1 Lingeling 587

The version of Lingeling submitted to the competition differs from the version submitted to the SAT Race 2010 described in [1] as follows. Similar to CryptoMiniSAT [7] our new garbage collection algorithm for reducing the number of learned clauses determines at run-time whether classic activity based heuristics or glues [2] are used. When a reduction operation is scheduled the average and standard deviation of the glue values of the still existing learned redundant clauses are computed. From this information the solvers tries to determine

https://doi.org/10.35011/fmvtr.2011-1

Technical Report 11/1, March 2011, FMV Reports Series
Institute for Formal Models and Verification, Johannes Kepler University
Altenbergerstr. 69, 4040 Linz, Austria

whether glues are useful for separating useful from useless learned clauses to be discarded. If the standard deviation is too small or too large we use activities. Since we only use four bits to represent glue values, a large average also shows that glues are not useful. So in some sense if the distribution does not fit nicely into our window of 15 glue values, we switch to activities (for *this* particular reduction). The actual glue used for these computations is actually scaled down by taking its square root. With less success, we also experimented with logarithmic scaling.

Our recent results on unhiding hidden tautologies and hidden literal removal [5] using the binary implication graph are incorporated as an additional *unhiding* inprocessing phase. Beside the *unhiding* preprocessor resp. inprocessor, since it is rescheduled in regular intervals and not just executed initially, we have also added a double lookahead procedure *lifting* for failed literal extraction and lifting of equivalences. Another recent technique [6, 4] for cheap reuse of the assignment stack during restarts has been integrated as well. This allows more frequent restarts, and we have reduced the base interval for Luby restarts to 10.

Finally, this version of Lingeling has partial support for incremental SAT solving and allows logging of API calls to a file, even though these two new features are not used in this competition.

2 Plingeling 587

In addition to sharing derived unit clauses [1] this new version of Plingeling also shares equivalences between different worker threads. The boss thread maintains a global table, which implements a union find data structure. Each worker thread imports and exports to this global table whenever a local union find data structure is completed and is about to be used for simplifying the thread local clause data base. This occurs in the already previously implemented *decompose* inprocessing phase, which uses strongly connected components of the binary implication graph, and also after lifting equivalences in the new *lifting* double lookahead procedure.

References

1. Lingeling, Plingeling, PicoSAT and PrecoSAT at SAT Race 2010. FMV Report Series Technical Report 10/1, Johannes Kepler University, Linz, Austria, 2010.
2. G. Audemard and L. Simon. Predicting learnt clauses quality in modern SAT solvers. In C. Boutilier, editor, *IJCAI*, pages 399–404, 2009.
3. A. Biere. Adaptive restart strategies for conflict driven sat solvers. In H. K. Büning and X. Zhao, editors, *SAT*, volume 4996 of *Lecture Notes in Computer Science*, pages 28–33. Springer, 2008.
4. M. Heule. About reusing the trail. Personal Communication.
5. M. Heule, M. Järvisalo, and A. Biere. Efficient CNF simplification based on binary implication graphs. Submitted.
6. A. Ramos, P. Van Der Tak, and M. Heule. Between restarts and backjumps. Submitted.
7. M. Soos. Cryptominisat 2.9.0.