Lingeling and Friends Entering the SAT Race 2015

Armin Biere
Institute for Formal Models and Verification
Johannes Kepler University Linz

Abstract—This is a solver description for our SAT solvers Lingeling, Treengeling and Plingeling entering the SAT Race 2015. We only focus on the difference to their 2014 versions. For further information we refer to previous solver descriptions [1], [2], [3], our POS’14 talk [4] and of course the source code.

Base Version

Two versions of Lingeling are submitted to the main sequential track of the SAT Race. The base version "baq" differs from the SAT competition 2014 version described in [3] as follows. The most important difference is due to adopting a variant of the Glucose restart scheme [5], which proved very effective on the SAT Competition 2014 benchmarks, particularly for small time limits, as our post-competition analysis showed.

A detailed evaluation of these changes related to restarts can be found in [6]. Further changes consist of adding a quaternary resolution inprocessor, similar to but much more conservative in adding quaternary resolvers than the previously existing ternary resolution component. The data structure for decision variable selection has been split into the original binary heap and an additional queue holding variables with a score of zero (initially or after rescoring). This reduces time spent in bumping variables etc. An extensive discussion of CDCL variable scoring schemes and their efficient implementation can be found in our SAT’15 conference paper [7].

Certain benchmarks in the SAT Competition 2014 turned out to have a ‘lucky phase’ solution, e.g., for instance can be solved by setting all variables to false. The previous version of Lingeling missed these simple solutions, since more advanced phase initialization heuristics force some variables to be set to true. This made these instances very hard to solve for Lingeling in contrast to all the MiniSAT and Glucose variants used in the competition, which find the solution without producing any conflict. Even though we consider the inclusion of these benchmarks in the 2014 competition set as highly problematic, we implemented a simple lucky phase checker which detects this situation and initializes phases accordingly.

Finally we added a poor man’s version of SAT sweeping. It uses similar refinement techniques as our more effective solution [8] based on blocked clause decomposition [8], [9], but it is easier to integrate and can be used during inprocessing [10]. Equivalent literal candidate pairs are checked in the same way as in [8]. However, if all variables in clauses containing those literals or their negations are assigned and propagated without leading to a conflict, this partial model is used for refining equivalence classes, while [8] uses full models.

Parameter Tuning

Version "baq" of Lingeling starts to explore techniques for automatic parameter selection in the spirit of portfolio solving, which explicitly was allowed in the SAT Race 2015. This is mainly motivated by results reported in [6], which clearly show that different restart intervals can have a huge impact on solving speed. Even though sophisticated dynamic restart schemes are useful in general, see again [6] for an extensive discussion, they still fail on certain instances, where either a faster or slower restart frequency would be much better. We observed, that these intervals often can be picked optimally for certain benchmark families, called buckets in [11].

Thus we added a simple classification scheme based on a k-nearest neighbor (KNN) algorithm, which predicts the origin of a benchmark (more precisely the SAT competition 2014 bucket). It uses the number of variables and clauses (of different length) of the original and simplified instance after one round of pre-/inprocessing. We generated optimized binary KNN classifiers using different k’s and features for each individual bucket. If the classifiers disagree the base version is used. Otherwise we pick the restart interval or other parameter settings determined to be best for the predicted bucket in separate experiments.

In these experiments with the base version "baq", we used the same set of parameter value choices as in Plingeling. To avoid exploring all combinations of parameter settings only one parameter setting was chosen to be different from the base version in each experiment, similar to how parameters are set for individual solver instances in Plingeling. Not all buckets lead to parameter changes, nor are all investigated parameter settings used.

Parameter tuning is disabled in the parallel front-ends Plingeling and Treengeling. However, Plingeling uses those new 13 different parameter settings for individual solver threads. Otherwise there is no difference compared to the 2014 version. The same applies to Treengeling. A failed attempt to make use of portfolio solving within a Cube-And-Conquer solver is included in the code but disabled.

License

The default license of Lingeling, Treengeling and Plingeling applies to these SAT Race 2015 versions as well. It allows the use of the software for academic, research and evaluation purposes. It further prohibits the use of the software in other competitions or similar events without explicit written permission. Please refer to the actual license, which comes with the source code, for more details, or contact the author.

Supported by FWF, NFN Grant S11408-N23 (RiSE).
REFERENCES


