The CryptoMiniSat 5 set of solvers at SAT Competition 2016

Mate Soos

I. Introduction

This paper presents the conflict-driven clause-learning SAT solver CryptoMiniSat v5 (*CMS5*) as submitted to SAT Competition 2016. CMS5 aims to be a modern, open-source SAT solver that allows for multi-threaded in-processing techniques while still retaining a strong CDCL component. In this description only the features relative to CMS4.4, the previous year's submission, are explained. Please refer to the previous years' description for details. In general, CMS5 is a in-processing SAT solver that usues optimized datastructures and finely-tuned timeouts to have good control over both memory and time usage of simplification steps.

A. Removal of uneeded code

Over the years, many lines of code has been added to CMS that in the end didn't help and often was detrimental to both maintinability and efficiency of the solver. Many such additions have now been removed. This simplifies understanding and developing the system. Further, it allows the system to be more lean especially in the tight loops such as propagation and conflict analysis where most of the time is spent.

B. Integration of ideas from COMiniSatPS

Some of the ideas from COMiniSatPS[1] have been included into CMS. In particular, the clause cleaning system employed and the switching restart have both made their way into CMS.

C. On-the-fly Gaussian Elimination

On-the-fly Gaussian elimination is again part of CryptoMiniSat. This is explicitly disabled for the compeititon, but the code is available and well-tested. This allows for special uses of the solver that other solvers, without on-the-fly Gaussian elimination, are not capable of.

D. Clause usefulness guessing

Besides glues and clause activites, CMS5 also tries to guess clause usefulness based on the trail size, the backjump level and the activity of the variables in the ancestor of the learnt clause. Although this is at a very early stage of development, it has been found to be helpful.

E. Auto-tuning

The version 'autotune' reconfigures itself after about 160K conflicts. The configuration picked is one of 2 different setups that vary many different parameters of the solving such as learnt clause removal strategy, restart strategy, and in-processing strategies. CMS5 was run on all SAT Comp'09 + 11 + 13 + 14 + 15 problems with both configurations, extracting relevant information from the all problems after they have been solved and simplified for 160K conflicts. configurations were then given to a machine learning algorithm (C5.0[2]) which built a decision tree from this data. This decision tree was then translated into C++ and compiled into the CMS5 source code.

References

[1] Oh, C.: MiniSat HACK 999ED, MiniSat HACK 1430ED and SWDiA5BY. In: SAT Competition 2014 Booklet. (201)

[2] Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1993)