

# Glucose and Syrup: Nine years in the SAT competitions

Gilles Audemard  
Univ. Lille-Nord de France  
CRIL/CNRS UMR8188  
audemard@cril.fr

Laurent Simon  
Bordeaux-INP / Univ. Bordeaux  
LaBRI/CNRS UMR 5800  
lsimon@labri.fr

**Abstract**—Glucose is a CDCL solver developed on top of Minisat nine years ago, with a special focus on removing useless clauses as soon as possible, and an original restart scheme based on the quality of recent learnt clauses. Syrup is the parallel version of Glucose, with an original lazy clauses exchanges policy, thanks to a one-watched scheme. We describe in this short solver description the small novelties introduced this year for the SAT 2018 competition.

## I. INTRODUCTION

Glucose is a CDCL (Conflict Driven Clause Learning) solver introduced in 2009 that tries to center all the components of the SAT solver around a measure of learnt clause quality, called LBD, for Literal Block Distance. This measure allows to delete a lot of learnt clauses from the beginning of the computation. From a practical point of view, it seems that this feature allows Glucose to produce more shorter proofs, which probably explains why Glucose and Syrup won a number of competitions in the last 9 years. A recent survey paper summarizes most of the improvements we added to the original Glucose [1]. Of course, the current short description does not mean to be exhaustive and the interested reader should refer to the previous paper.

In a few words, however, Glucose enters SAT competitions/races [2], [3] every years since its creation. Glucose is based on the internal architecture of minisat [4] (especially for the VSIDS implementation, the 2-Watched scheme and the memory management of clauses (garbage collection, ...)). It is based on the notion of Literal Block Distance, as aforementioned, a measure that is able to estimate the quality of learnt clauses [5]. This measure simply counts the number of distinct decision levels of literals occurring in learnt clauses, at the time of their creation. Thanks to that, new strategies for deleting clauses were proposed. Moreover, the solver constantly watch the quality of the last learnt clauses and triggers a restart when the quality is worst than the average. Recent developments includes a way of postponing restarts when the number of assigned literals suddenly increases without conflicts (a SAT solution may be then expected). In the last version of Glucose, a special strategy allowed the solver to decide which strategy to use with respect to a set of identified extreme case [6].

Indeed, learnt clauses removal, restarts, small modifications of the VSIDS heuristic are based on the concept of LBD. The core engine of Glucose (and Syrup) is 8 years old. Syrup is a major improvement of Glucose on which we focused most of our efforts in the last years.

## II. NEW COMPONENTS

The 2018 version of Glucose and Syrup are very similar to the 2016 ones, with two improvements. The main modifications are based on the extension of the recent LCM strategies proposed last year [7] (which "revived" the vivification technique [8]). We observed that the LCM strategy was not always performed on clauses of small LBD only, because LCM was not triggered right after clause database reduction, and thus the order of clauses traversed by the LCM was not based on a sorted order of learnt clauses. However, we observed that LCM was more efficient when not always run on good clauses only (LCM can replace clauses, and thus may delete a good clause). We observed that LCM was more efficient when active clauses were kept, in addition to clauses of small LBD. Glucose is now keeping 10% of the most active clauses in addition to the usual LBD based ranking. In addition, we integrated the LCM technique into Syrup (good clauses found during the LCM reduction were not shared initially). The integration into Syrup was not obvious because LCM seemed to be much more efficient on sequential solvers only. LCM was thus not activated on all the cores in the parallel version.

## III. PARALLEL VERSION OF GLUCOSE

We used a version with 24 and 48 cores this year for the parallel versions of Glucose (called Glucose-Syrup).

## IV. ALGORITHM AND IMPLEMENTATION DETAILS

Glucose uses a special data structure for binary clauses, and a very limited self-subsumption reduction with binary clauses, when the learnt clause is of interesting LBD. The parallel version uses a special data structure for sharing clauses that may prevent some clause to be shared when it is full.

## V. ACKNOWLEDGMENTS

Recent developments of Glucose-Syrup is supported by the French Project SATAS ANR-15-CE40-0017. We would also like to thanks the organizers of the competition for their efforts.

## REFERENCES

- [1] G. Audemard and L. Simon, "On the glucose sat solver," vol. 27, no. 01.
- [2] —, "Glucose: a solver that predicts learnt clauses quality," *SAT Competition*, pp. 7–8, 2009.
- [3] —, "Glucose 2.3 in the sat 2013 competition," *Proceedings of SAT Competition*, pp. 42–43, 2013.

- [4] N. Eén and N. Sörensson, “An extensible SAT-solver,” in *SAT*, 2003, pp. 502–518.
- [5] G. Audemard and L. Simon, “Predicting learnt clauses quality in modern sat solvers,” in *IJCAI*, 2009.
- [6] —, “Extreme Cases in SAT,” in *19th International Conference on Theory and Applications of Satisfiability Testing(SAT’13)*, 2013, p. To appear.
- [7] M. Luo, C. Li, F. Xiao, F. Manyà, and Z. Lü, “An effective learnt clause minimization approach for CDCL SAT solvers,” in *Proceedings of IJCAI 2017*, 2017, pp. 703–711.
- [8] C. Piette, Y. Hamadi, and L. Sais, “Vivifying propositional clausal formulae,” in *ECAI 2008 - 18th European Conference on Artificial Intelligence*, 2008, pp. 525–529.