NaPS: Nagoya pseudo-Boolean solver

Masahiko Sakai† and Hidetomo Nabeshima‡

† Nagoya University
‡ University of Yamanashi

1 Introduction

NaPS (Nagoya pseudo-Boolean solver) is a Pseudo-Boolean solver branched from MiniSAT+ version 1.0 [4]. NaPS accepts the Dimacs PB-instances for PB-competitions having no non-linear constraints. The extended features of the NaPS input language are listed as follows:

- Negated Boolean variables: negations `~x` of variables `x` are allowed in constraints.
- Each constraint may include an implication: constraints are in the following form:

  `d` *literal implication-operator linear-constraint* `;`

  where, `=>`, `<=`, and `<=>` are allowed as an *implication-operator*.

NaPS optionally accepts Dimacs format for cnf/wcnf and has MaxSat mode. It has a (possibly partial) model-count facility.

2 Techniques

2.1 NaPS-PB'16

NaPS-PB'16 is version 1.02b5, fixed some bugs of 1.02b2 submitted to PB'16 competition. The solver transforms PB-instances to ordinary Boolean SAT-instances mainly via reduced ordered BDDs with two-clause coding [1]. It sometimes preprocesses coefficients by a multi-based decomposition [3]. It constructs a single BDD for a PB-constraint with equation [6]. In optimization, it adopts the alternative strategy [6], which combines sequential and binary strategies, where the latter is with a ratio of $1 : 2$.

As an underlying SAT-solver, it uses GlueMiniSat 2.2.6-ucore [5].

2.2 NaPS

NaPS is with version 1.03. The main differences from 1.02b5 are listed as follows:

- It uses Kissat [2] version 3.1.1 for decision problems and CaDiCaL [2] version 2.0.0 for others by default.
- Goal expression is divided into several fragments when possible and is optimized in multi-stage mode.

2.3 NaPS-GM

NaPS-GM is with version 1.03 but uses GlueMiniSat 2.2.6-ucore [5].

References

1. I. Abío, R. Nieuwenhuis, A. Oliveras, E. Rodríguez-Carbonell, and V. Mayer-Eichberger. A new look at BDDs for Pseudo-Boolean constraints. *Journal of Artificial Intelligence Research* (JAIR), vol. 45, pp. 443–480, 2012.
2. Armin Biere, Mathias Fleury and Florian Pollitt:˙CaDiCaL vivinst, IsaSAT, Gimsatul, Kissat, and TabularaSAT Entering the SAT Competition 2023. In *Proc. of SAT Competition 2023 – Solver, Benchmark and Proof Checker Descriptions*, vol. B-2023-1 of Department of Computer Science Publicatoins B, pp. 14–15, University of Helsinki, 2023.
3. M. Codish, Y. Fekete, C. Fuhs, and P. Schneider-Kamp. Optimal base encodings for Pseudo-Boolean constraints. In *TACAS*, volume 6605 of *LNCS*, pages 189–204, 2011.
4. N. Eén and N. Sörensson. Translating Pseudo-Boolean constraints into SAT. *Journal on Satisfiability, Boolean Modeling and Computation* (JSAT), vol. 2, no. 1–4, pp. 1–26, 2006.
5. H. Nabeshima, K. Iwanuma, and K. Inoue. On-the-fly lazy clause simplification based on binary resolvents. In *ICTAI*, IEEE, pp. 987–995, 2013.
6. M. Sakai and H. Nabeshima. Construction of an ROBDD for a PB-constraint in band form and related techniques for PB-solvers. *IEICE Transaction on Information and Systems*, vol. E98-D, no. 6, pp. 1121–1127, 2015.