Fun-sCOP

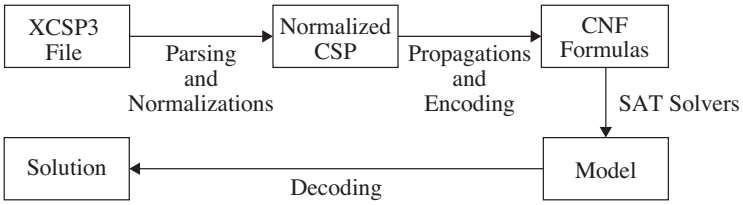
XCSP3 Competition 2019

Takehide Soh[1], Daniel Le Berre[2], Hidetomo Nabeshima[3], Mutsunori Banbara[4], and Naoyuki Tamura[1]

[1] Kobe University, Japan, {soh@lion.,tamura@}kobe-u.ac.jp
[2] CRIL-CNRS, Université d'Artois, France, leberre@cril.fr
[3] University of Yamanashi, Japan, nabesima@yamanashi.ac.jp
[4] Nagoya University, Japan, banbara@nagoya-u.jp

1 Overview

Fun-sCOP is a SAT-based constraint programming system written in Scala, which aims to be a re-implementation of Sugar [5] written in Java. Compared to the previous version named scop, Fun-sCOP equips the hybrid encoding integrating the order and log encodings [3]. The following figure shows the framework of Fun-sCOP. We explain each part of this framework in the remaining of this paper.



2 Parsing and Normalizations

Parsing is done by using the official tool XCSP3-Java-Tools [5]. Fun-sCOP accepts constraints in the XCSP3-core language[6].

Normalizations are executed as follows:

- **Global Constraints** are decomposed into intensional constraints. We use extra pigeon hole constraints [5] for alldifferent constraints.
- **Extensional constraints** are translated into intensional constraints by using a variant of multi-valued decision diagrams. This is a difference to ones in Sugar.
- **Intensional Constraints** are normalized to be in the form of a conjunction of disjunctions of linear comparisons $\sum_i a_i x_i \geq k$ where a_i's are integer coefficients, x_i's are integer variables and k is an integer constant. Tseitin transformation is used to avoid the combinatorial explosion.

[5] https://github.com/xcsp3team/XCSP3-Java-Tools
[6] http://www.xcsp.org/specifications

3 Propagations and Encoding Methods

Constraint propagations are executed to the normalized CSP (clausal CSP, i.e., in the form of CNF over linear comparisons $\sum_i a_i x_i \geq k$) to remove redundant values, variables, and linear comparisons. It is done by using an AC3 like algorithm.

Encoding methods are as follows:

- **Order Encoding** [7, 6] is an encoding method using propositional variables $p_{x \geq d}$'s meaning $x \geq d$ for each domain value d of each integer variable x. To encode linear comparisons, Algorithm 1 of the literature [6] is used in Fun-sCOP.
- **Hybrid Encoding** [3] is an encoding method integrating the order and log encodings without channeling constraints. In the hybrid encoding, each variable is encoded by either the order encoding or the log encoding, and each constraint is encoded according to its variables' encoding. The degree of hybridization can be controlled by classifying the order-encoded and log-encoded variables. Fun-sCOP uses the criterion *domain product* to classify variables as same as in [3].

4 SAT Solvers

GlueMiniSat [2] is used for the order encoding. It is a winning solver in the SAT solver competitions, and the submitted version of GlueMiniSat uses a special propagator for the axiom clauses of the order encoding.

CryptoMiniSat [4] is used for the hybrid encoding. It is also a winning solver in the recent SAT solver competitions, and shows a good performance in our preliminary evaluation.

ManyGlucose [1] is used for parallel CSP track. It is a deterministic parallel SAT solver based on Glucose.

References

1. Gotou, Y., Nabeshima, H.: ManyGlucose. In: Proceedings of SAT Competition 2018: Solver and Benchmark Descriptions, volume B-2018-1 of Department of Computer Science Series of Publications B, University of Helsinki. p. 27 (2018)
2. Nabeshima, H., Iwanuma, K., Inoue, K.: GlueMiniSat 2.2.8. In: Proceedings of SAT Competition 2014. pp. 35–36 (2014)
3. Soh, T., Banbara, M., Tamura, N.: Proposal and evaluation of hybrid encoding of CSP to SAT integrating order and log encodings. International Journal on Artificial Intelligence Tools 26(1), 1–29 (2017)
4. Soos, M., Nohl, K., Castelluccia, C.: Extending SAT solvers to cryptographic problems. In: Proceedings of the 12th International Conference on Theory and Applications of Satisfiability Testing (SAT 2009), LNCS 5584. pp. 244–257 (2009)
5. Tamura, N., Banbara, M.: Sugar: a CSP to SAT translator based on order encoding. In: Proceedings of the 2nd International CSP Solver Competition. pp. 65–69 (2008)
6. Tamura, N., Banbara, M., Soh, T.: PBSugar: Compiling pseudo-boolean constraints to SAT with order encoding. In: Proceedings of the 25th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2013). pp. 1020–1027 (Nov 2013)
7. Tamura, N., Taga, A., Kitagawa, S., Banbara, M.: Compiling finite linear CSP into SAT. Constraints 14(2), 254–272 (2009)