Duck, Gregory J.

SMCHR: satisfiability modulo constraint handling rules. (English) Zbl 1260.68059
Theory Pract. Log. Program. 12, No. 4-5, 601-618 (2012).

Summary: Constraint handling rules (CHRs) are a high-level rule-based programming language for specification and implementation of constraint solvers. CHR manipulates a global store representing a flat conjunction of constraints. By default, CHR does not support goals with a more complex propositional structure including disjunction, negation, etc., or CHR relies on the host system to provide such features.

In this paper we introduce satisfiability modulo constraint handling rules (SMCHR): a tight integration of CHR with a modern Boolean satisfiability (SAT) solver for quantifier-free formulae with an arbitrary propositional structure. SMCHR is essentially a satisfiability modulo theories (SMT) solver where the theory T is implemented in CHR. The execution algorithm of SMCHR is based on lazy clause generation, where a new clause for the SAT solver is generated whenever a rule is applied. We shall also explore the practical aspects of building an SMCHR system, including extending a "built-in" constraint solver supporting equality with unification and justifications.

MSC:

68N17 Logic programming

Keywords:

CHR; satisfiability modulo theories; lazy clause generation

Software:

SMCHR

Full Text: DOI

References:

[1] Een, Proceedings of the Sixth International Conference on Theory and Applications of Satisfiability Testing (2003)

[2] DOI: 10.1007/978-3-540-92243-8_3 · Zbl 1229.68025 · doi:10.1007/978-3-540-92243-8_3

[3] Duck, International Conference on Logic Programming pp 90– (2004) · doi:10.1007/978-3-540-27775-0_7

[4] DOI: 10.1017/S1471068411000494 · Zbl 1244.68023 · doi:10.1017/S1471068411000494

[5] Holzbaur, Proceedings of 1992 International Symposium on Programming Language Implementation and Logic Programming pp 260– (1992)

[6] DOI: 10.1016/S0743-1066(96)00065-9 · Zbl 0877.68020 · doi:10.1016/S0743-1066(96)00065-9

[7] DOI: 10.1007/10704567_7 · doi:10.1007/10704567_7

[8] DOI: 10.1017/S1471068409990123 · Zbl 1186.68096 · doi:10.1017/S1471068409990123

[9] Sarna-Starosta, Proceedings of the 5 pp 3– (2008)

[10] DOI: 10.1007/s10601-008-9064-x · Zbl 1192.68654 · doi:10.1007/s10601-008-9064-x

[11] Demoen, Proceedings of the Sixteenth International Conference on Logic Programming pp 260– (1999)

[12] DOI: 10.1145/1217856.1217859 · Zbl 1326.68164 · doi:10.1145/1217856.1217859

[13] DOI: 10.1145/1995376.1995394 · doi:10.1145/1995376.1995394

[14] DOI: 10.1002/spe.4380180902 · doi:10.1002/spe.4380180902