

SMT-COMP'05

System Description: Sammy

Michael DeCoster George Hagen
Cesare Tinelli Hantao Zhang

Computer Science Department
The University of Iowa
Iowa City, IA, USA
{decoster,ghagen,tinelli,hzhang}@cs.uiowa.edu

Sammy is an SMT checker written in OCaml and C based on the $DPLL(T)$ architecture [GHN⁺04]. This architecture defines a DPPL-style engine for propositional solving, $DPLL(X)$, parametrized (in X) by a solver for a theory of interest. A $DPLL(T)$ system is obtained by plugging in a solver for a background theory T conforming to a certain interface. The solver is in essence a ground decision procedure for T , although it must provide additional functionalities than simple yes-no answers (see [GHN⁺04] for details).

By and large, Sammy is the implementation of the $DPLL(X)$ engine. The theory solver used in this version of Sammy is the combined decision procedure of the CVC Lite prover [BB04]. This procedure decides the satisfiability of set of literals modulo a background theory which is a rich combination of several background theories commonly used in SMT applications. CVC Lite is written in a highly modular way, with many of its core functionalities provided as libraries. Through a selection of CVC Lite's library functions one can carve out of CVC Lite a theory solver for the $DPLL(T)$ architecture, which is what is done in this case. Sammy's $DPLL(X)$ engine relies on a *satbox* derived from the SATO propositional solver [Zha97]. The *satbox* module is responsible for such $DPLL$ operation as unit propagation, (propositional) lemma learning, and backjumping.

The $DPLL(T)$ architecture accepts sets of ground CNF formulas in the signature of the theory T possibly expanded with free function or predicate symbols. Sammy on the other hand accepts arbitrary first-order formulas, in both the CVC Lite and in the SMT-LIB format [RT05].

Quantified, non-CNF formulas are dealt with by an additional wrapper around Sammy's DPLL(T) subsystem. In essence, this wrapper generates ground clauses from an input formula as needed, and feeds them into the DPLL(T) subsystem. At any time, the current set of generated ground clauses is logically entailed by the input formula. The clause generation feature is based on new (incomplete) heuristics to be described in forthcoming work.

Sammy's overall algorithm is sound, complete and terminating for the satisfiability of ground (equivalently, existential) formulas modulo the combined background theory. It is however only sound for arbitrary first-order formulas.

SMT-COMP divisions: QF_UF, QF_RDL, QF_IDL, QF_LRA, QF_LIA, QF_UFIDL, QF_AUFLIA.

References

- [BB04] Clark W. Barrett and Sergey Berezin. Cvc lite: A new implementation of the cooperating validity checker category b. In *Proceedings of the 16th International Conference on Computer Aided Verification, CAV 2004, Boston, MA, USA*, volume 3114 of *Lecture Notes in Computer Science*, pages 515–518. Springer, 2004.
- [GHN⁺04] Harald Ganzinger, George Hagen, Robert Nieuwenhuis, Albert Oliveras, and Cesare Tinelli. DPLL(T): Fast decision procedures. In R. Alur and D. Peled, editors, *Proceedings of the 16th International Conference on Computer Aided Verification, CAV'04 (Boston, Massachusetts)*, volume 3114 of *Lecture Notes in Computer Science*, pages 175–188. Springer, 2004.
- [RT05] Silvio Ranise and Cesare Tinelli. The smt-lib standard: Version 1.1. Technical report, INRIA-Lorraine and The University of Iowa, April 2005.
- [Zha97] H. Zhang. SATO: An efficient propositional prover. In William McCune, editor, *Proceedings of the 14th International Conference on Automated deduction*, volume 1249 of *Lecture Notes in Artificial Intelligence*, pages 272–275, Berlin, July 1997. Springer.