Congruence Closure with Free Variables

Haniel Barbosa, Pascal Fontaine, and Andrew Reynolds

Satisfiability Modulo Theories (SMT)

- The problem of assessing whether a FOL formula has a model consistent with background theories
- CDCL($\mathcal{T}$) framework solves SMT by combining SAT and theory solvers
- Quantifier reasoning generally through *heuristic instantiation* based on E-matching

Contributions

A unifying framework for instantiating quantified formulas with equality and uninterpreted functions [Barbosa, Fontaine, Reynolds. TACAS'17]

- Formalizing underlying problem for instantiation in SMT
- Lifting congruence closure to accommodate free variables
- Casting existing instantiation techniques in framework
- Techniques for efficient implementation

E-ground (dis)unification

Framework is based on the problem of E-ground (dis)unification

Definition: *Given conjunctive sets of equality literals E and L, with E ground, finding a substitution σ s.t. $E \models L\sigma$*

- Solution space can be restricted into ground terms from $E \cup L$
- NP-complete
  - ▷ NP: solutions can checked in polynomial time
  - ▷ NP-hard: reduction of 3-SAT into the entailment
- Variant of classic (non-simultaneous) rigid E-unification

$$s_1\sigma \simeq t_1\sigma, \ldots, s_n\sigma \simeq t_n\sigma \models u\sigma \simeq v\sigma$$

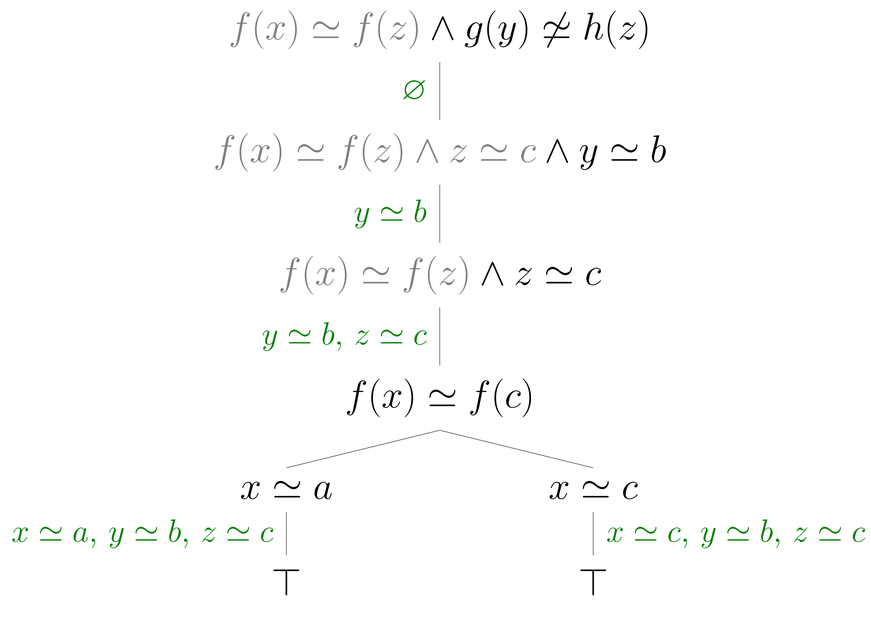
Congruence Closure with Free Variables (CCFV)

A sound, complete and terminating calculus for solving E-ground (dis)unification

- Search for solutions as a series of AND-OR constraints depending on the entailment of conditions of literals in L
- Congruence closure as a core element
  - ▷ All terms inferred equal are kept in the same class
  - ▷ Constraints to be entailed are normalized according to partial solutions
- Different possibilities for building solutions are handled with branching and backtracking

Finding solutions σ for $E \models L\sigma$:

$$E \models L\sigma$$
$$f(a) \simeq f(c) \wedge g(b) \not\simeq h(c) \models (f(x) \simeq f(z) \wedge g(y) \not\simeq h(z))\,\sigma$$



Existing instantiation techniques as special cases

▷ Conflict-based instantiation [RTM14]
- ⊕ CCFV provides formal guarantees and more clear extensions

▷ E-matching based heuristic instantiation [DNS05; MB07]
- ⊕ CCFV allows to easily discard instances already entailed by E

▷ Model-based instantiation [GM09; RTG+13]
- ⊕ No need for a secondary ground SMT solver
- ⊕ No need to guess solutions

Implementation techniques

- Model minimisation
- Top symbol indexing of E-graph from ground congruence closure
- Selection strategies

$$E \models f(x, y) \simeq h(z) \wedge x \simeq t \wedge \ldots$$

- Eagerly checking whether constraints can be discarded

Experiments and Conclusions

- CCFV has been implemented in the SMT solvers veriT and CVC4
- Techniques based on CCFV:
  - **t** : trigger instantiation through CCFV;
  - **c** : conflict based instantiation through CCFV;
  - **b** : breadth-first version of CCFV rather than the depth-first one;
  - **e** : eagerly discarding branches with unmatchable applications;
  - **d** : discards already entailed trigger based instances
- Comparison of instantiation based SMT solvers

Logic	Class	Z3	**cvc+d**	**cvc+e**	**cvc**	**verit+tc**	**verit+tcb**	**verit+t**	**verit**
UF	grasshopper	418	411	420	415	430	**435**	418	413
	sledgehammer	1249	1438	**1456**	1428	1277	1278	1134	1066
UFIDL	all	**62**	**62**	**62**	**62**	58	58	58	58
UFLIA	boogie	**852**	844	834	801	706	690	660	661
	sexpr	**26**	12	11	11	7	7	5	5
	grasshopper	341	322	326	319	356	**361**	340	335
	sledgehammer	1581	1944	**1953**	1929	1790	1799	1620	1569
	simplify	**831**	766	706	705	803	801	735	690
	simplify2	**2337**	2330	2292	2286	2307	2303	2291	2177
	Total	7697	**8129**	8060	7956	7734	7736	7261	6916

veriT: + 800 out of 1 785 unsolved problems

CVC4: + 200 out of 745 unsolved problems

Benchmarks in the "UF", "UFLIA", "UFLRA" and "UFIDL" categories of SMT-LIB, which have 8,701 benchmarks annotated as *unsatisfiable* that are not trivially solved by all systems. Timeout is 30s.

References

[BFR17] Haniel Barbosa, Pascal Fontaine, and Andrew Reynolds. "Congruence Closure with Free Variables". In: Tools and Algorithms for Construction and Analysis of Systems (TACAS). Ed. by Axel Legay and Tiziana Margaria. Vol. 10206. Lecture Notes in Computer Science. 2017, pp. 214–230.

[DNS05] David Detlefs, Greg Nelson, and James B. Saxe. "Simplify: A Theorem Prover for Program Checking". In: J. ACM 52.3 (2005), pp. 365–473.

[GM09] Yeting Ge and Leonardo de Moura. "Complete Instantiation for Quantified Formulas in Satisfiabiliby Modulo Theories". In: Computer Aided Verification (CAV). Ed. by Ahmed Bouajjani and Oded Maler. Vol. 5643. Lecture Notes in Computer Science. Springer, 2009, pp. 306–320.

[MB07] Leonardo de Moura and Nikolaj Bjørner. "Efficient E-Matching for SMT Solvers". In: Proc. Conference on Automated Deduction (CADE). Ed. by Frank Pfenning. Vol. 4603. Lecture Notes in Computer Science. Springer, 2007, pp. 183–198.

[RTG+13] Andrew Reynolds, Cesare Tinelli, Amit Goel, Sava Krsti, Morgan Deters, and Clark Barrett. "Quantifier Instantiation Techniques for Finite Model Finding in SMT". In: Proc. Conference on Automated Deduction (CADE). Ed. by Maria Paola Bonacina. Vol. 7898. Lecture Notes in Computer Science. Springer, 2013, pp. 377–391.

[RTM14] Andrew Reynolds, Cesare Tinelli, and Leonardo Mendonça de Moura. "Finding conflicting instances of quantified formulas in SMT". In: Formal Methods In Computer-Aided Design (FMCAD). IEEE, 2014, pp. 195–202.