iProver v3.9 (SMT-COMP-2024)

Konstantin Korovin

The University of Manchester, UK
`konstantin.korovin@manchester.ac.uk`

iProver[1] [2,11] is a theorem prover for quantified first-order logic with support for arithmetical reasoning. For quantified reasoning iProver interleaves instantiation calculus Inst-Gen [6,12] with ordered resolution and superposition calculi [19]. First-order clauses are exchanged between calculi during the proof search. Clauses generated during the proof search are grounded and submitted into SAT and SMT solvers for ground reasoning. In turn, instantiations in the Inst-Gen calculus are guided by propositional models. The reasoning method behind iProver is refutationally complete for pure first-order logic with equality.

iProver is implemented in OCaml. It natively accepts typed first-order formulas with arithmetic TFF0 [20] in CNF, with extensions it accepts SMT2, TPTP FOF/TFF0, AIG, QBF among other formats. In SMT iProver supports all combinations of quantifiers, uninterpreted functions, arrays, data types, linear and non-linear arithmetic. iProver supports arbitrary precision arithmetic. iProver integrates Z3 [17] and MiniSAT [5] for ground reasoning during the reasoning phase and uses Vampire [15] for clausification and axiomatisations of theories [18] which is done as the initial problem transformation into the TFF0 format.

Key components of iProver include:

- Preprocessing including [7]: predicate elimination [10], splitting, semantic filtering, subtyping and definition elimination.
- Inst-Gen calculus [6,12]: model guided incremental instantiation with unification, dismatching constraints and global subsumption [11,12].
- Resolution with dynamic literal selection, ordering restrictions and simplifications forward/backward: subsumption, global subsumption and subsumption resolution.
- Superposition with simplifications [2]: forward/backward: demodulation, light normalisation, subsumption, global subsumption and subsumption resolution. A range of indexes are implemented for inference rules and simplifications including perfect and non-perfect discrimination trees, feature vector indexes. Simplifications and indexes are partitioned within iProver's given clause algorithm into active, passive and immediate sets.
- AC reasoning: AC joinability and AC normalisation [3].
- Ground joinability, connectedness, encompassment demodulation [4]
- All calculi above operate on a combination of priority queues for selecting next formulas for inferences. These priorities can be specified in the command line as a lexicographic combination of different formula characteristics.

---

[1] iProver is available at: https://gitlab.com/korovin/iprover

- Abstraction-refinement loop interleaving under and over approximations of first-order formulas [16] on top of the calculi above.
- Finite model finding for first-order logic via translation into the EPR fragment [1] and non-cyclic sorts [13].
- Proof and model reconstruction, which is non-trivial due simplifications involving the global search state such as global subsumption [14].
- Heuristics selection and scheduling which specify over 100 prover parameters governing simplifications and interleaving of the calculi. The heuristics are learnt using machine learning HOS-ML framework which is based on Bayesian hyper parameter optimisation and dynamic clustering [8, 9]. The heuristic learning process is syntax independent and we reused heuristics trained on TPTP problems (rather than SMT) which should be unbiased towards SMT problems. A parallel 4-core schedule was precomputed from the learnt heuristics using constraint solving, maximising the schedule coverage of solved training problems for the given time limit [8].

References

1. Baumgartner, P., Fuchs, A., de Nivelle, H., Tinelli, C.: Computing finite models by reduction to function-free clause logic. J. Appl. Log. **7**(1), 58–74 (2009)
2. Duarte, A., Korovin, K.: Implementing superposition in iProver (system description). In: Peltier, N., Sofronie-Stokkermans, V. (eds.) Automated Reasoning - 10th International Joint Conference, IJCAR 2020, Paris, France, July 1-4, 2020, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12167, pp. 388–397. Springer (2020)
3. Duarte, A., Korovin, K.: AC simplifications and closure redundancies in the superposition calculus. In: Das, A., Negri, S. (eds.) The 30th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX'21). Lecture Notes in Computer Science, Springer (2021), to appear
4. Duarte, A., Korovin, K.: Ground joinability and connectedness in the superposition calculus. In: Blanchette, J., Kovács, L., Pattinson, D. (eds.) Automated Reasoning - 11th International Joint Conference, IJCAR 2022, Haifa, Israel, August 8-10, 2022, Proceedings. Lecture Notes in Computer Science, vol. 13385, pp. 169–187. Springer (2022)
5. Eén, N., Sörensson, N.: An extensible sat-solver. In: Giunchiglia, E., Tacchella, A. (eds.) Theory and Applications of Satisfiability Testing, 6th International Conference, SAT 2003. Santa Margherita Ligure, Italy, May 5-8, 2003 Selected Revised Papers. Lecture Notes in Computer Science, vol. 2919, pp. 502–518. Springer (2003)
6. Ganzinger, H., Korovin, K.: New directions in instantiation-based theorem proving. In: 18th IEEE Symposium on Logic in Computer Science (LICS 2003), 22-25 June 2003, Ottawa, Canada, Proceedings. pp. 55–64. IEEE Computer Society (2003)
7. Hoder, K., Khasidashvili, Z., Korovin, K., Voronkov, A.: Preprocessing techniques for first-order clausification. In: Cabodi, G., Singh, S. (eds.) Formal Methods in Computer-Aided Design, FMCAD 2012, Cambridge, UK, October 22-25, 2012. pp. 44–51. IEEE (2012)
8. Holden, E.K., Korovin, K.: Heterogeneous heuristic optimisation and scheduling for first-order theorem proving. In: Kamareddine, F., Coen, C.S. (eds.) The 14th Conference on Intelligent Computer Mathematics (CICM 2021). Lecture Notes in Computer Science, Springer (2021), to appear

9. Holden E., K., Korovin, K.: SMAC and XGBoost your theorem prover. In: Hales, T., Kaliszyk, C., Kumar, R., Schulz, S., Urban, J. (eds.) 4th Conference on Artificial Intelligence and Theorem Proving. pp. 93–96. AITP (2019)
10. Khasidashvili, Z., Korovin, K.: Predicate elimination for preprocessing in first-order theorem proving. In: Creignou, N., Berre, D.L. (eds.) Theory and Applications of Satisfiability Testing - SAT 2016 - 19th International Conference, Bordeaux, France, July 5-8, 2016, Proceedings. Lecture Notes in Computer Science, vol. 9710, pp. 361–372. Springer (2016)
11. Korovin, K.: iProver – an instantiation-based theorem prover for first-order logic (system description). In: Armando, A., Baumgartner, P., Dowek, G. (eds.) Proceedings of the 4th International Joint Conference on Automated Reasoning, (IJCAR 2008). Lecture Notes in Computer Science, vol. 5195, pp. 292–298. Springer (2008)
12. Korovin, K.: Inst-Gen — A Modular Approach to Instantiation-Based Automated Reasoning. In: Voronkov, A., Weidenbach, C. (eds.) Programming Logics, vol. 7797, pp. 239–270. Springer Berlin Heidelberg (2013)
13. Korovin, K.: Non-cyclic sorts for first-order satisfiability. In: Fontaine, P., Ringeissen, C., Schmidt, R.A. (eds.) Frontiers of Combining Systems - 9th International Symposium, FroCoS 2013, Nancy, France, September 18-20, 2013. Proceedings. Lecture Notes in Computer Science, vol. 8152, pp. 214–228. Springer (2013)
14. Korovin, K., Sticksel, C.: A note on model representation and proof extraction in the first-order instantiation-based calculus Inst-Gen. In: Schmidt, R.A., Papacchini, F. (eds.) Proceedings of the 19th Automated Reasoning Workshop (ARW'12). pp. 11–12. School of Computer Science, The University of Manchester, Manchester (2012)
15. Kovács, L., Voronkov, A.: First-order theorem proving and vampire. In: Sharygina, N., Veith, H. (eds.) Computer Aided Verification - 25th International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings. Lecture Notes in Computer Science, vol. 8044, pp. 1–35. Springer (2013)
16. López-Hernández, J.C., Korovin, K.: An abstraction-refinement framework for reasoning with large theories. In: Galmiche, D., Schulz, S., Sebastiani, R. (eds.) Automated Reasoning - 9th International Joint Conference, IJCAR 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 14-17, 2018, Proceedings. Lecture Notes in Computer Science, vol. 10900, pp. 663–679. Springer (2018)
17. de Moura, L.M., Bjørner, N.: Z3: an efficient SMT solver. In: Ramakrishnan, C.R., Rehof, J. (eds.) Tools and Algorithms for the Construction and Analysis of Systems, 14th International Conference, TACAS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008, Budapest, Hungary, March 29-April 6, 2008. Proceedings. Lecture Notes in Computer Science, vol. 4963, pp. 337–340. Springer (2008)
18. Reger, G., Suda, M.: Set of support for theory reasoning. In: Eiter, T., Sands, D., Sutcliffe, G., Voronkov, A. (eds.) IWIL Workshop and LPAR Short Presentations. Kalpa Publications in Computing, vol. 1, pp. 124–134. EasyChair (2017)
19. Robinson, J.A., Voronkov, A.: Handbook of Automated Reasoning (in 2 volumes). Elsevier and MIT Press (2001)
20. Sutcliffe, G., Schulz, S., Claessen, K., Baumgartner, P.: The TPTP typed first-order form with arithmetic. In: Bjørner, N., Voronkov, A. (eds.) Logic for Programming, Artificial Intelligence, and Reasoning - 18th International Conference, LPAR-18, Mérida, Venezuela, March 11-15, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7180, pp. 406–419. Springer (2012)