Towards a Jitting VM for Prolog Execution

Carl Friedrich Bolz, Michael Leuschel, and David Schneider

Lehrstuhl Softwaretechnik und Programmiersprachen
Institut für Informatik, Universität Düsseldorf, Germany

Abstract. Most Prolog implementations are implemented in low-level languages such as C and are based on a variation of the WAM instruction set [5], which enhances their performance but makes them hard to write. We present a high-level continuation-based [3] Prolog interpreter written in RPython, a restricted subset of Python [1]. This interpreter is annotated with hints, so that it can be fed through the PyPy tracing JIT generator, which incorporates partial evaluation techniques [2]. The resulting Prolog implementation is surprisingly efficient: it clearly outperforms existing implementations of Prolog in high-level languages like Java [4]. Moreover, on some benchmarks, our system outperforms state-of-the-art WAM-based Prolog implementations. The talk tries to show that PyPy can indeed form the basis for implementing programming languages other than Python. Furthermore, we believe that our results showcase the great potential of the tracing JIT approach for logic programming languages like Prolog.[1]

References

1. D. Ancona, M. Ancona, A. Cuni, and N. D. Matsakis. RPython: a step towards reconciling dynamically and statically typed OO languages. In *Proceedings of the 2007 symposium on Dynamic languages*, pages 53–64, Montreal, Quebec, Canada, 2007. ACM.
2. C. F. Bolz, A. Cuni, M. Fijakowski, and A. Rigo. Tracing the meta-level: PyPy's tracing JIT compiler. In *Proceedings of the 4th workshop on the Implementation, Compilation, Optimization of Object-Oriented Languages and Programming Systems*, pages 18–25, Genova, Italy, 2009. ACM.
3. T. Lindgren. A Continuation-Passing style for Prolog. In *Symposium on Logic Programming*, pages 603–617, 1994.
4. G. Piancastelli, A. Benini, A. Omicini, and A. Ricci. The architecture and design of a malleable object-oriented prolog engine. In *Proceedings of the 2008 ACM symposium on Applied computing*, pages 191–197, Fortaleza, Ceara, Brazil, 2008. ACM.
5. P. van Roy. 1983-1993: The wonder years of sequential Prolog implementation. *Journal of Logic Programming*, 19:385–441, 1994.

[1] This research is partially supported by the BMBF funded project PyJIT (nr. 01QE0913B; Eureka Eurostars).