

# Analyses, Optimizations and Extensions of Constraint Handling Rules: Ph.D. Summary

Tom Schrijvers\*

Dept. of Computer Science, K.U.Leuven, Belgium

## Abstract

This is a summary of the Ph.D. thesis of Tom Schrijvers [4].

Constraint Handling Rules (CHR) [3] is a rule-based language commonly embedded in a host language. It combines elements of Constraint Logic Programming and term rewriting. Several implementations of CHR exist: in Prolog, Haskell, Java and HAL. Typical applications of CHR are in the area of constraint solving, but currently CHR is also used in a wider range of applications, such as type checking, natural language processing and multi-agent systems.

In this work we contribute program analyses, program optimizations and extensions of the CHR language.

For the optimized compilation of CHR we present several new optimizations: code specialization for ground constraints, anti-monotonic delay avoidance, hashtable constraint stores and a new late storage optimization. These and other optimizations have been implemented in a new state-of-the-art CHR system: the K.U.Leuven CHR system [5] which is currently available in SWI-Prolog [10], XSB [9] and hProlog [2].

Abstract interpretation is a general technique for program analysis [1]. We propose a framework of abstract interpretation for the CHR language [7], in particular for the formulation of analyses that drive program optimization. This framework allows for the uniform formulation of program analyses as well as easier improvements and combinations of existing analyses. We also evaluate analyses for theoretical properties, confluence and time complexity, on a practical case study to investigate their relevance.

We contribute two extensions to the expressivity of CHR. The first extension comprises the integration of CHR with tabled execution [8]. Tabled execution avoids many forms of non-termination and is useful for automatic program optimization through the dynamic reuse of previous computations. The second extension automatically provides implication checking functionality to constraint solvers written in CHR [6]. Implication checking is an essential building block for formulating complex constraints in terms of basic constraints and for composing constraint solvers.

---

\* Research Assistant of the Fund for Scientific Research - Flanders (Belgium)(F.W.O. - Vlaanderen)

## Acknowledgements

I would like to thank my supervisor Bart Demoen, the jury members of my doctoral committee and all my co-authors for their help and support.

## References

1. Patrick Cousot and Radhia Cousot. Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fix-points. In *POPL '77: Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*, pages 238–252, Los Angeles, California, 1977. ACM Press.
2. Bart Demoen. hProlog. <http://www.cs.kuleuven.ac.be/~bmd/hProlog/>.
3. Thom Frühwirth. Theory and practice of constraint handling rules. *Journal of Logic Programming*, 37(1–3):95–138, October 1998.
4. Tom Schrijvers. *Analyses, Optimizations and Extensions of Constraint Handling Rules*. PhD thesis, K.U.Leuven, Leuven, Belgium, June 2005.
5. Tom Schrijvers and Bart Demoen. The K.U.Leuven CHR system: Implementation and application. In Thom Frühwirth and Marc Meister, editors, *First Workshop on Constraint Handling Rules: Selected Contributions*, pages 1–5, Ulm, Germany, May 2004.
6. Tom Schrijvers, Bart Demoen, Gregory Duck, Peter Stuckey, and Thom Frühwirth. Automatic Implication Checking for CHR Constraints. In Horatiu Cirstea and Narciso Martí-Oliet, editors, *RULE'05: Proceedings of the 6th International Workshop on Rule-Based Programming*, Nara, Japan, April 2005.
7. Tom Schrijvers, Peter Stuckey, and Gregory Duck. Abstract Interpretation for Constraint Handling Rules. In *PPDP'05: Proceedings of the 7th International Symposium on Principles and Practice of Declarative Programming*, Lisbon, Portugal, July 2005. ACM Press.
8. Tom Schrijvers and David S. Warren. Constraint handling rules and tabled execution. In Bart Demoen and Vladimir Lifschitz, editors, *ICLP'04: Proceedings of the 20th International Conference on Logic Programming*, volume 3132 of *Lecture Notes in Computer Science*, pages 120–136, St-Malo, France, September 2004. Springer Verlag.
9. David S. Warren et al. The XSB Programmer's Manual: version 2.7, vols. 1 and 2, January 2005. <http://xsb.sf.net>.
10. Jan Wielemaker. SWI-Prolog release 5.4.0, 2004. <http://www.swi-prolog.org/>.