COLIBRI

Bruno Marre Benjamin Blanc Patricia Mouy
Zakaria Chihani Franck Vedrine François Bobot

June 2, 2019

COLIBRI is a library (COnstraint LIBrary for veRIfication) developed at CEA LIST and used for verification or test data generation purposes since 2000, using the techniques of constraint programming. The variety of types and constraints provided by COLIBRI makes it possible to use it in many testing tools at CEA LIST like GATeL [3], for model based testing from Lustre/SCADE, and Osmose [1], for structural testing from binary code, PathCrawler tool for concolic C testing.

COLIBRI is a finite domain solver that use usual constraint programming techniques. The basis is that one or many domains are attached to each terms, and the constraints tighten the domains of one term using the domains of the other terms involve in the constraint. When no constraints could improve anymore the domain of any term, a decision is made. Usually it splits the domain in two, but other techniques can be used as 3B filtering which decide on the extremes of the domains and if unsatisfiable try to improve this bound.

The domains are very specific, no reduction to simpler theory like bit-blasting is used. COLIBRI uses a domain of union of intervals for real and integer inter-reduced with a domain of congruence. For floating points it uses a domain of intervals. For bit-vector a domain that indicates if the bits are set, unset or unknown.

In addition to these domains which reason on local properties, COLIBRI uses for integer, reals, and floating point DBM. The DBM use lots of patterns to be able to do limited but useful non-linear reasoning.

The power of the reasoning of COLIBRI, in particular in the floating point, are due to the information sharing and inter dependencies of all these reasoning techniques.

The combination of all the components of COLIBRI is simplified since, like all CP-solver, all the domains and constraints are improving the same model.

COLIBRI gained last year the ability to handle SMTLIB2 format (`QF_FP`, and `QF_BVFP` theory), but since historically it handled only simple and double format with nearest to even rounding[4, 2], the reasoning for other rounding mode is minimal (except in useful case like truncation, ceiling, ...), and other formats are not supported.

Last year COLIBRI gained the ability to reason about uninterpreted functions and arrays. This year the work as been mainly done to support new versions of ECLiPSe Prolog [5] in which COLIBRI is implemented.

References

[1] S. Bardin and P. Herrmann. Structural testing of executables. In *the First International Conference on Software Testing, Verification, and Validation (ICST 2008)*, pages 22–31. IEEE Computer Society, 2008.

[2] Z. Chihani, B. Marre, F. Bobot, and S. Bardin. Sharpening Constraint Programming approaches for Bit-Vector Theory. In *CPAIOR 2017. International Conference on AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, Integration of AI and OR Techniques in Constraint Programming, Padova, Italy, June 2017.

[3] B. Marre and B. Blanc. Test selection strategies for Lustre descriptions in GATeL. *Electronic Notes in Theoretical Computer Science*, 111:93–111, 2005.

[4] B. Marre, F. Bobot, and Z. Chihani. Real Behavior of Floating Point Numbers. In *The SMT Workshop*, Heidelberg, Germany, July 2017. SMT 2017, 15th International Workshop on Satisfiability Modulo Theories.

[5] J. Schimpf and K. Shen. ECLiPSe - from LP to CLP. *Theory and Practice of Logic Programming*, 12(1-2):127–156, 2011.