Approximate Quantified Constraint Solving
User Manual — Version 0.5

Stefan Ratschan

May 10, 2002

1 Introduction

Solving quantified constraints has applications in numerous fields (e.g., control theory [1, 14, 13, 7], stability analysis [16, 22, 17, 23, 11, 10], electrical Engineering [24], and others [6, 26, 18, 2, 12, 9]). However, classical methods for their solution [25, 4, 5] have various drawbacks:

- They are too slow for big examples,
- they often generate huge output, and
- they do not provide partial information before computing the total result.

This software package is based on an algorithm [21] that try to solve the above problems by computing approximate solutions up to a user-provided error bound. It uses various sub-algorithms from the area of interval computation [19, 20, 15] and constraint programming [8].

2 Input Syntax

Quantified constraints can contain the following symbols:

Quantifiers: `FORALL`, `EXISTS`

Connectives: `~`, `\/`, `/\`, `==>`, `<==>`, `not`, `and`, `or`, `implies`, `iff`

Predicates: `<`, `<=`, `>`, `>=`

Functions: `*`, `+`

Constants: Integers, floating point intervals

Variables:

All alphanumeric symbols are prefix and use functional notation (i.e., arguments in parenthesis), the other symbols are infix. Quantifiers have three arguments:

1. A list of quantified variables
2. A floating point bounding box on these variables
3. The quantified constraint

Although the software also allows equality predicates `=` it currently does not produce satisfying answers. Work on constraints with equalities is in progress.

3 Produced Output

The output is a set of boxes on which the input is guaranteed to be true, a set of boxes, on which the input is guaranteed to be false, and a set of boxes for which we do not know anything. For graphical output the true boxes are green and the false boxes red.

4 Running AQCS

After starting the program, it asks for an input constraint, the free (i.e., not quantified variables), and their range. Example inputs are contained in the distribution package.

For controlling the solver further, one can use arguments when calling the program. The most important are:

- -d: Provides graphical output (only for less then three free variables)
- -e EPS: Allows the unknown area of the result to be a fraction of EPS of the total area.
- -h: Lists further possible arguments.

5 Further Reading

The basic algorithm is described in a theoretical paper [21]. Some preprints are available from the author's webpage (`http://www.risc.uni-linz.ac.at/people/sratscha`).

References

[1] C. Abdallah, P. Dorato, W. Yang, R. Liska, and S. Steinberg. Applications of quantifier elimination theory to control system design. In *4th IEEE Mediterranean Symposium on Control and Automation*, Crete, Greece, 1996.

[2] F. Benhamou and F. Goualard. Universally quantified interval constraints. In *Proc. of the Sixth Intl. Conf. on Principles and Practice of Constraint Programming (CP'2000)*, number 1894 in LNCS, Singapore, 2000. Springer Verlag.

[3] B. F. Caviness and J. R. Johnson, editors. *Quantifier Elimination and Cylindrical Algebraic Decomposition*. Springer, 1998.

[4] G. E. Collins. Quantifier elimination for the elementary theory of real closed fields by cylindrical algebraic decomposition. In *Second GI Conf. Automata Theory and Formal Languages*, volume 33 of *LNCS*, pages 134–183. Springer Verlag, 1975. Also in [3].

[5] G. E. Collins and H. Hong. Partial cylindrical algebraic decomposition for quantifier elimination. *Journal of Symbolic Computation*, 12:299–328, 1991. Also in [3].

[6] A. Dolzmann, T. Sturm, and V. Weispfenning. Real quantifier elimination in practice. In B. H. Matzat, G.-M. Greuel, and G. Hiss, editors, *Algorithmic Algebra and Number Theory*, pages 221–248. Springer, 1998.

[7] P. Dorato, W. Yang, and C. Abdallah. Robust multi-objective feedback design by quantifier elimination. *Journal of Symbolic Computation*, 24:153–159, 1997.

[8] P. V. Hentenryck, D. McAllester, and D. Kapur. Solving polynomial systems using a branch and prune approach. *SIAM Journal on Numerical Analysis*, 34(2), 1997.

[9] H. Hong. Non-linear constraints solving over real numbers in constraint logic programming. Technical Report 92-08, RISC-Linz, 1992.

[10] H. Hong, R. Liska, and S. Steinberg. Logic, quantifiers, computer algebra and stability. *SIAM News*, 30(6):10–,13, 1997.

[11] H. Hong, R. Liska, and S. Steinberg. Testing stability by quantifier elimination. *Journal of Symbolic Computation*, 24(2):161–187, 1997.

[12] N. I. Ioakimidis. Quantifier elimination in applied mechanics problems with cylindrical algebraic decomposition. *International Journal of Solids and Structures*, 34(30):4037–4070, 1997.

[13] M. Jirstrand. Nonlinear control system design by quantifier elimination. *Journal of Symbolic Computation*, 24(2):137–152, 1997.

[14] M. Jirstrand. *Algebraic Methods for Inequality Constraints in Control*. PhD thesis, Linköping University, Department of Electrical Engineering, Division of Automatic Control, 1998. No. 527.

[15] R. B. Kearfott. Interval computations: Introduction, uses, and resources. *Euromath Bulletin*, 2(1):95–112, 1996.

[16] R. Liska and S. Steinberg. Applying quantifier elimination to stability analysis of difference schemes. *Computer Journal*, 36(5):497–593, 1993.

[17] R. Liska and S. Steinberg. Solving stability problems using quantifier elimination. In R. Jeltsch and M. Mansour, editors, *Stability Theory, Proceedings of Conference on Centennial Hurwitz on Stability Theory*, pages 205–210, Basel, 1996. Birkhäuser Verlag. International Series of Numerical Mathematics (ISNM) Vol. 121.

[18] R. Loos and V. Weispfenning. Applying linear quantifier elimination. *The Computer Journal*, 36(5):450–462, 1993.

[19] R. E. Moore. *Interval Analysis*. Prentice Hall, Englewood Cliffs, NJ, 1966.

[20] A. Neumaier. *Interval Methods for Systems of Equations*. Cambridge Univ. Press, Cambridge, 1990.

[21] S. Ratschan. Continuous first-order constraint satisfaction. In *Proceedings of Artificial Intelligence and Symbolic Computation*, LNCS. Springer, 2002. To appear, `http://www.risc.uni-linz.ac.at/people/sratscha/preprints.html`.

[22] S. Steinberg and R. Liska. Stability analysis by quantifier elimination. In G. Jacob, N. Ousous, and S. Steinberg, editors, *Proceedings SC 93, International IMACS Symposium on Symbolic Computation, New Trends and Developments*, pages 62–67, 1993. Lille, France, June 14 - 17, 1993.

[23] S. Steinberg and R. Liska. Stability analysis by quantifier elimination. *Mathematics and Computers in Simulation*, 42(4-6):629–638, 1996.

[24] T. Sturm. Reasoning over networks by symbolic methods. *Applicable Algebra in Engineering Communication and Computing*, 10(1):79–96, September 1999.

[25] A. Tarski. *A Decision Method for Elementary Algebra and Geometry*. Univ. of California Press, Berkeley, 1951. Also in [3].

[26] V. Weispfenning. Simulation and optimization by quantifier elimination. *Journal of Symbolic Computation*, 24(2), 1997.